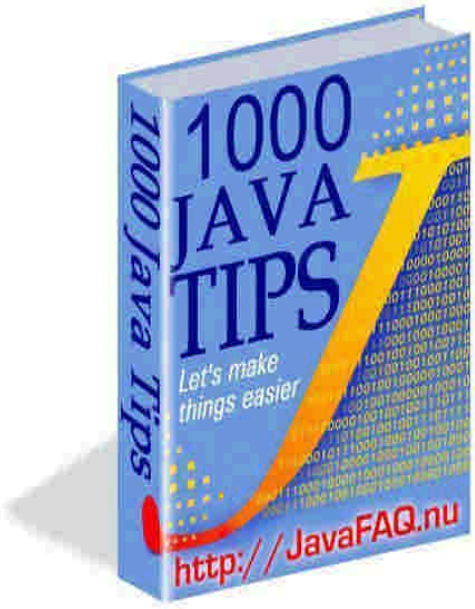
?!! javaFAQ.nu
JavaQ:A

JAVA!
JavaA.com



Copyright Info

This book can be copied and distributed in any form. Nobody, except for copyright owners, has a right to modify the book content. It can be published online on any site if this copyright info is displayed as well. All the tips can be used in other publications or published separately if this copyright info is added.
This book is completely free and nobody has a right to charge for this book, its distribution and so on.

Copyright owners keep the right on mistake that could be made in the book and have no responsibility for possible consequences of such error.
The copyright owners of this book are:
Alexandre Patchine – http://JavaA.com , http://JavaFAQ.nu and
Dr. Heinz M. Kabutz in "More Java Tips by Dr. Heinz M. Kabutz" section - http://www.javaspecialists.co.za

You can contribute to this book if you have valuable Java advice, tip, and code example and so on. Your copyright info – name, site will be added as well.

Reviews

"This book has saved me hours of research while coding Java. From the obvious to the obscure, this book answers the real world questions that bring development to a halt. And the authors really know their stuff. A great help."

- *Robin Kay* from user opinions on CNET about free first edition

"My 5.5 years Java experience can to be improved even more!"

- *Net Rambler from Europe* from user opinions on CNET about free first edition

"Nice format to learn Java. Excellent e-book for developers who do not have mentors around in the initial stages."

From user opinions on CNET about free first edition

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Foreword

Hello dear reader!

"1000 Java Tips" is my collection of tricky questions and answers from my newsletters for last four years and answers I gave to people in my emails.

Let me ask you a few questions:

1. Are you preparing for Java certification exam and want to be sure that you will pass it from first attempt? This book gives you real examples how to use Java.

2. Are you looking for a job as a Java programmer and want to be prepared for the Java interview questions?

3. Are you looking for a good design idea?

You have come to right place!

This e-Book will give you more knowledge, better perspective to programming and computer science especially. It covers wide range of questions.

My "1000 Java Tips" e-book will help you to understand Java much better and be well prepared for your Java Certification Examination.

The e-Book is made in the form of questions and answers. I included also the big section of Advanced Java Tips by Dr. Heinz Kabutz where topics are bigger. They are well written articles! Every article exploring one area and gives you examples.

This e-book is an easy-to-navigate huge collection of Java Tips.

Easy style, right proportion of humor and Java lets you read them at once, in one breath!

Author of “1000 Java Tips” e-book,
Alexandre Patchine.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Table of Contents

Copyright Info 2
Reviews 2
Foreword 3
Table of Contents 4
1. Introduction 7
2. Applets 8
3. Databases & Beans 25
4. Distributed systems 32
5. File Systems 34
6. Graphics, AWT, Swing 42
7. General Java 66
8. Java Bugs 175
9. Java Hardware 176
10. Java Tools and Software 177
11. Job, fun 193
12. Miscellaneous 196
13. Mobile Java 218
14. Networking, JSP, Mail... etc 230
15. Operation Systems and Java 251
16. RMI 260
17. Security 265
18. Servlets, Servers, Mail etc 269
19. Sound, Speech and Multimedia 283
20. String, text, numbers, I/O 290
21. Threads 316
22. Code Examples 332
23. More Java Tips by Dr. Heinz M. Kabutz 346
.....[Issue 001] - Deadlocks 346
.....[Issue 002] - Anonymous Inner Classes 349
.....[Issue 003] - Logging part 1 353
.....[Issue 004] - Logging part 2 357
.....[Issue 005] - Dynamic Proxies - Short Tutorial 361
.....[Issue 006] - Implementation code inside interfaces 367
.....[Issue 007] - java.awt.EventQueue 369
.....[Issue 008] - boolean comparisons 374
.....[Issue 009] - Depth-first Polymorphism 377
.....[Issue 010] - Writing GUI Layout Managers 381
.....[Issue 011] - Hooking into the shutdown call 387
.....[Issue 012] - Setting focus to second component of modal dialog 390
.....[Issue 013a] - Serializing GUI Components Across Network 394
.....[Issue 013b] - Follow-up 400
.....[Issue 014] - Insane Strings 403
.....[Issue 015] - Implementing a SoftReference based HashMap 407
.....[Issue 016] - Blocking Queue 412
.....[Issue 017a] - Switching on Object Handles 416
.....[Issue 017b] - Follow-up 420
.....[Issue 018] - Class names don't identify a class 425
.....[Issue 019] - Finding Lost Frames 429
.....[Issue 020] - Serializing Objects Into Database 433
.....[Issue 021] - Non-virtual Methods in Java 437
.....[Issue 022] - Classloaders Revisited: "Hotdeploy" 440

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

.....[Issue 023] - Socket Wheel to handle many clients 455
.....[Issue 024] - Self-tuning FIFO Queues 461
.....[Issue 025] - Final Newsletter..... 468
.....[Issue 026] - Package Versioning 475
.....[Issue 027] - Circular Array List..... 479
.....[Issue 028] - Multicasting in Java 487
.....[Issue 029] - Determining Memory Usage in Java 494
.....[Issue 030] - What do you Prefer?..... 501
.....[Issue 031] - Hash, hash, away it goes!..... 508
.....[Issue 032] - Exceptional Constructors - Resurrecting the dead 513
.....[Issue 033] - Making Exceptions Unchecked 518
.....[Issue 034] - Generic Types with Dynamic Decorators 523
.....[Issue 035] - Doclets Find Bad Code 532
.....[Issue 036] - Using Unicode Variable Names..... 536
.....[Issue 037] - Checking that your classpath is valid..... 541
.....[Issue 038a] - Counting Objects Clandestinely 546
.....[Issue 038b] - Counting Objects Clandestinely - Follow-up..... 552
.....[Issue 039] - Why I don't read your code comments 554
.....[Issue 040] - Visiting your Collection's Elements..... 557
.....[Issue 041] - Placing components on each other..... 563
.....[Issue 042] - Speed-kings of inverting booleans 567
.....[Issue 043] - Arrgh, someone wants to kill me!..... 571
.....[Issue 044] - Review: Object-Oriented Implementation of Numerical Methods 576
.....[Issue 045] - Multi-line cells in the JTable 582
.....[Issue 046] - "The compiler team is writing useless code again ..."..... 586
.....[Issue 047] - Lack of Streaming leads to Screaming..... 589
.....[Issue 048] - Review: The Secrets of Consulting..... 595
.....[Issue 049] - Doclet for finding missing comments..... 599
.....[Issue 050] - Commenting out your code?..... 607
.....[Issue 051] - Java Import Statement Cleanup..... 610
.....[Issue 052] - J2EE Singleton 617
.....[Issue 053] - Charting unknown waters in JDK 1.4 Part I..... 627
.....[Issue 054] - HashMap requires a better hashCode() - JDK 1.4 Part II 631
.....[Issue 054b] - Follow-up to JDK 1.4 HashMap hashCode() mystery 639
.....[Issue 055] - Once upon an Oak 640
.....[Issue 056] - Shutting down threads cleanly..... 646
.....[Issue 057] - A Tribute to my Dad, Hans Rudolf Kabutz..... 652
.....[Issue 058] - Counting bytes on Sockets..... 655
.....[Issue 059] - When arguments get out of hand..... 666
.....[Issue 059b] - Follow-up to Loooong Strings..... 675
.....[Issue 060] - Nulling variables and garbage collection..... 677
.....[Issue 061] - Double-checked locking..... 681
.....[Issue 062] - The link to the outer class..... 684
.....[Issue 062b] - Follow-up and Happy New Year! 689
.....[Issue 063] - Revisiting Stack Trace Decoding..... 691
.....[Issue 064] - Disassembling Java Classes..... 697
.....[Issue 065] - Wait, Cursor, Wait!..... 699
.....[Issue 066] - Book Review: Java Performance Tuning by Jack Shirazi 708
.....[Issue 067] - BASIC Java..... 714
.....[Issue 068] - Appending Strings..... 719
.....[Issue 069] - Treating Types Equally - or - Life's Not Fair! 725
.....[Issue 069b] - Results of last survey..... 730
.....[Issue 070] - Too many dimensions are bad for you..... 731
.....[Issue 070b] - Multi-Dimensional Arrays - Creation Performance..... 734
.....[Issue 071] - Overloading considered Harmful..... 736

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

.....[Issue 072] - Java and Dilbert 743
.....[Issue 073] - LinkedHashMap is Actually Quite Useful 746
.....[Issue 074] - GoF Factory Method in writing GUIs 751
.....[Issue 075] - An Automatic Wait Cursor: WaitCursorEventQueue 761
.....[Issue 076] - Asserting Locks 790
.....[Issue 077] - "Wonderfully disgusting hack" 796
.....[Issue 078] - com.maxoft.memory.MemoryCounter for Java 1.4 798
.....[Issue 079] - Generic toString() 806
.....[Issue 080] - Many Public Classes in One File 814
.....[Issue 081] - Catching Exceptions in GUI Code 816
.....[Issue 082] - TristateCheckBox based on the Swing JCheckBox 822
.....[Issue 083] - End of Year Puzzle 828
.....[Issue 083b] - End of Year Puzzle Follow-up 830
.....[Issue 084] - Ego Tripping with Webservices 831
.....[Issue 085] - Book Review: Pragmatic Programmer 839
.....[Issue 086] - Initialising Fields before Superconstructor call 844
.....[Issue 086b] - Initialising Fields before Superconstructor call (Follow-up) 848
24. XML 850
25. About The Author 852

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

1. Introduction

Do not lose time here, please go ahead and read my book right now!

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

2. Applets

- **Question:** What are restrictions for an applet? What are applets prevented from doing?

  Answer: In general, applets loaded over the net are prevented from reading and writing files on the client file system, and from making network connections except to the originating host.

  In addition, applets loaded over the net are prevented from starting other programs on the client. Applets loaded over the net are also not allowed to load libraries, or to define native method calls. If an applet could define native method calls, which would give the applet direct access to the underlying computer.

- **Question:** How can my applet require some optional packages (specific vendor and versions)?

  Answer: Since version 1.3 Java introduced support for an expanded set of Jar-file manifest attributes.

  The following portion of a sample manifest file illustrates the attributes that are available for use by optional packages.

  ```
  Extension-Name: javax.help
  Specification-Vendor: Sun Microsystems, Inc
  Specification-Version: 1.0
  Implementation-Vendor-Id: com.sun
  Implementation-Vendor: Sun Microsystems, Inc
  Implementation-Version: 1.0
  ```

  Note that all of the above manifest attributes should be included in an optional package's manifest. If the plug-in looks for but doesn't find one of these attributes in an optional package's manifest, it will assume that the optional package is not suitable for the applet.

  Note: Optional packages are the new name for what used to be known as standard extensions. The "extension mechanism" is that functionality of the Java 2 SDK and Java 2 Runtime Environment that supports the use of optional packages.

  Optional packages are packages of Java classes (and any associated native code) that application developers can use to extend the functionality of the core platform. The extension mechanism allows the Java virtual machine (VM) to use the classes of

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

the optional extension in much the same way as the VM uses classes in the Java 2 Platform. The extension mechanism also provides a way for needed optional packages to be retrieved from specified URLs when they are not already installed in the Java 2 SDK or Runtime Environment.

From SUN's description of the Java Extension Mechanism

- **Good to know:** Invoking modal popup dialogs from an applet's stop or destroy methods is not recommended if you use Java 1.4.1 and below. Using modal popup dialogs inside these methods could result in problematic behavior, including NullPointerException, the dialog box not being displayed, the dialog box remaining alive even after the applet was destroyed, or in the worst case, browser hang. This problem has been fixed in 1.4.2.

- **Question:** Do I need special server software to use applets?

  Answer: No. Java applets may be served by any HTTP server. On the server side they are handled the same as any other file, such as a text, image, or sound file. All the special action happens when the applet class files are interpreted on the client side by a Java technology-enabled browser, such as HotJava browser or 1.x or Netscape 3.x/4.x.
  source: http://java.sun.com/products/jdk/faq.html#A8

- **Question:** I know that applets have limited possibility to do many things. It is about network connections, file reading/writhing and more.
  Can applet read all system properties and if not how many of them are restricted?

  Answer: Applets can read quite many of system properties by using:
  String ss = System.getProperty(String key):

  java.version
  java.vendor
  java.vendor.url
  java.class.version
  os.name
  os.arch
  os.version
  file.separator
  path.separator
  line.separator

  Applets are prevented from reading these system properties:
  java.home
  java.class.path
  user.name
  user.home
  user.dir
  source: http://java.sun.com/sfaq/

- **Question:** I have an applet. I have successfully added the class to a JAR, but when the web page tries to load the applet, the browser gives us an error "can't find class file". When unzipping the JAR, my added class is present.

  Answer: Do not forget to add the "ARCHIVE" option to the APPLET tag like this

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
<applet code=my.class archive=archive.jar width=10 height=10>
</applet>
```

If "archive" is missing the browser will not know that your class is in a jar file...

- **Question:** Is there a way to write information to cookies through an Applet?

  Answer: Applets are not allowed to access information on the Client Side...it overrides the Sandbox restriction...

  However, JavaScript can access cookies and Java applet can talk to JavaScript.

  So, a Java applet indirectly has access to cookies... An example of JavaScript <--> Java was published in my newsletter before...

  Look for the JSObject class from Netscape which works in IE as well.

- **Question:** I am writing a chat applet and I would like to put icons in user messages. Is it possible?

  Answer: A text area can parse HTML. Add image (jpg or gif) into html code and it will display your icons.

- **Question:** Can one applet load another applet on runtime, and can these two applets talk?

  Answer: the Answer found on "Internet related technologies" site:

  "Yes one applet can load another applet. And since you need both the applet to communicate, you need to have both applets in the same page. But the problem in this case is that both the applets will load independently of each other. You can control the behaviour of the other applet from the first one. But your requirement is of loading another applet from the first one.

  You can have two frames in a page. In one frame there will be your first applet which will load automatically when the page is loaded. And the other frame will contain a blank page. Now on certain event in the first applet you can use "this.showDocument("URL") " to make a call to another applet which you want to load. Give the target parameter as the other frame. (You will have to give the name of the other frame) Now when these two applets are loaded they can exchange data in their life time."

  http://developer.irt.org/script/4074.htm

- **Question:** How do I hide system properties that applets are allowed to read by default?

  Answer: There's no way to hide the ten system properties (see below) from applets loaded into a Java-enabled browser. The reason is that the browsers don't consult any external files as part their Java configuration, as a security precaution, including the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

~/.hotjava/properties file.

From the appletviewer, you can prevent applets from finding out anything about your system by redefining the property in your

~/.hotjava/properties file. For example, to hide the name of the operating system that you are using, add this line to your

~/.hotjava/properties file:

os.name=null

Ten OS propertis that accessible by applet:
java.version
java.vendor
java.vendor.url
java.class.version
os.name
os.arch
os.version
file.separator
path.separator
line.separator
-

From Java Applet Security FAQ

Question: The Applet tag has many attributes. Some of them are mandatory and some are optional. Could you list them? Part 1

Answer:

The attributes of the applet tag give the browser information about the applet. The only mandatory attributes are CODE (or OBJECT), WIDTH, and HEIGHT. Each attribute is described below.

CODEBASE = codebaseURL

This OPTIONAL attribute specifies the base URL of the applet--the directory that contains the applet's code. If this attribute is not specified, then the document's URL is used.

ARCHIVE = archiveList

This OPTIONAL attribute describes one or more archives containing classes and other resources that will be "preloaded". The classes are loaded using an instance of an AppletClassLoader with the given CODEBASE.

The archives in archiveList are separated by ",". NB: in JDK1.1, multiple APPLET tags with the same CODEBASE share the same instance of a ClassLoader. This is used by some client code to implement inter-applet communication. Future JDKs may provide other mechanisms for inter-applet communication. For security reasons, the applet's class loader can read only from the same codebase from which the applet was

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

started. This means that archives in archiveList must be in the same directory as, or in a subdirectory of, the codebase. Entries in archiveList of the form ../a/b.jar will not work unless explicitly allowed for in the security policy file (except in the case of an http codebase, where archives in archiveList must be from the same host as the codebase, but can have ".."'s in their paths.)

CODE = appletFile

This REQUIRED attribute gives the name of the file that contains the applet's compiled Applet subclass. This file is relative to the base URL of the applet. It cannot be absolute. One of CODE or OBJECT must be present. The value appletFile can be of the form classname.class or of the form packagename.classname.class.

OBJECT = serializedApplet

This attribute gives the name of the file that contains a serialized representation of an Applet. The Applet will be deserialized. The init() method will *not* be invoked; but its start() method will. Attributes valid when the original object was serialized are *not* restored. Any attributes passed to this APPLET instance will be available to the Applet; we advocate very strong restraint in using this feature. An applet should be stopped before it is serialized. One of CODE or OBJECT must be present.

ALT = alternateText

Question: The Applet tag has many attributes. Some of them are mandatory and some are optional. Could you list them? Part 2

This OPTIONAL attribute specifies any text that should be displayed if the browser understands the APPLET tag but can't run Java applets.

NAME = appletInstanceName

This OPTIONAL attribute specifies a name for the applet instance, which makes it possible for applets on the same page to find (and communicate with) each other.

WIDTH = pixels HEIGHT = pixels

These REQUIRED attributes give the initial width and height (in pixels) of the applet display area, not counting any windows or dialogs that the applet brings up.

ALIGN = alignment

This OPTIONAL attribute specifies the alignment of the applet. The possible values of this attribute are the same as those for the IMG tag: left, right, top, texttop, middle, absmiddle, baseline, bottom, absbottom.

VSPACE = pixels HSPACE = pixels

These OPTIONAL attributes specify the number of pixels above and below the applet (VSPACE) and on each side of the applet (HSPACE). They're treated the same way as the IMG tag's VSPACE and HSPACE attributes.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

<PARAM NAME = appletAttribute1 VALUE = value>

<PARAM NAME = appletAttribute2 VALUE = value> . . .

This tag is the only way to specify an applet-specific attribute. Applets access their attributes with the getParameter() method.

- **Question:** What is the difference between applets loaded over the net and applets loaded via the file system?

  Answer: There are two different ways that applets are loaded by a Java system. The way an applet enters the system affects what it is allowed to do.

  If an applet is loaded over the net, then it is loaded by the applet class loader, and is subject to the restrictions enforced by the applet security manager.

  If an applet resides on the client's local disk, and in a directory that is on the client's CLASSPATH, then it is loaded by the file system loader. The most important differences are applets loaded via the file system are allowed to read and write files applets loaded via the file system are allowed to load libraries on the client applets loaded via the file system are allowed to exec processes applets loaded via the file system are allowed to exit the virtual machine applets loaded via the file system are not passed through the byte code verifier Java-enabled browsers use the applet class loader to load applets specified with file: URLs. So, the restrictions and protections that accrue from the class loader and its associated security manager are now in effect for applets loaded via file: URLs.

  This means that if you specify the URL like so:

  Location: file:/home/me/public_html/something.html

  and the file something.html contains an applet, the browser loads it using its applet class loader.
  -

  From Java Applet Security FAQ

- **Question:** Every time I access an Applet browser downloads all related classes.

  I would like to use the cashed copy if they classes were not modified and do not load them again and again...

  Answer: Please use the tag:

  <PARAM NAME="cache_archive" VALUE="xx.jar,yy.jar,zz.jar">

- **Question:** How do I increase the level of debugging information displayed in the Java Console of Netscape?

  Answer: When the Java Console is active hit any number between 0-9, 0 = No Information, 9 = Maximum Information

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** My applets are big and it takes much time to download (to browser) when a network is slow. Is there any way to decrease the start up time?

  Answer: The good solution is to make a small core .jar file that contains part applet that is necessary to start it up (initial GUI elements) and rest of classes can be loaded later when applet is already running. Nobody use 100 % of functionality directly and will give an impression that everything is ready... :-)

  This solution is good due to another reason also. If browser lost connection during loading of one big .jar file you need to start everything from scratch. If it happens with our solution browser will cash part of classes and it will save time for reloading!

  Usually you don't need all the functionality of the applet at once, so the applet can operate before all of its possible functionality is downloaded. And .jar and .class files does get cached by most browsers.

- **Question:** I write my first applet and it become very huge! It is an applet but looks like huge Java Application.
  Could you point me what is most is important for having a small applet?

  Answer:
  1. Use compiler optimization: javac -O But check it the size anyway. Sometime it makes the code bigger.

  2. Use jar files instead of class files

  3. Try to use inheritance as much as possible: than more code you can reuse than less new lines you have to add.

  4. Try to use standard APIs. Often they are better optimized in size than some private exotic packages. Of course often they have better methods and so on but try to use efficiently what we have already!

  5. Use short names.

  6. Do not initialize big arrays because. They will be initialized and put directly into bytecode. You can do it later on the fly.

- **Question:** Why do I get message like "wrong magic number" when I am trying to run applet? What is a magic number?

  Answer: The first thing a JVM does when it loads a class is check that the first four bytes are (in hex) CA FE BA BE. This is the "magic number" and that's why you are getting that error, you are trying to load a file that isn't a class and so the class loader in the JVM is throwing out that exception.

  Make sure you transfer the class files to site in binary mode, rather than text or ASCII mode.
  An error from the browser saying "cannot start applet ... bad magic number" usually means that one of the class files on the server is corrupted. '
  Replace your class binary files on the web server; clean up the cache of your browser, and reload your applet.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I've got problems with the Socket class (network)
I've got problems with the Socket class. I use it inside an applet (I've written a small chat box). I have code like this:

  Socket s = new Socket("192.168.0.4", 13780);

  When the server I'm connecting to is on the same machine as the client, it works. When the server is an other machine, both NS and IE give an error message like:

  Security: Can't connect to 192.168.0.4 with origin ''

  Does anyone know how I can fix this?

  Answer: The standard security concept for an applet is the 'sandbox'. An applet can't talk outside it's memory space, can't talk to any files at all, and cannot talk to anything on the internet except the same machine that it's 'parent' HTML page originated from. So your applet can never talk to 192.168.0.4 unless the HTML came from 192.168.0.4

- **Question:** How do I view the error output from my Java applets in IE?

  Answer: The file windows\Java\Javalog.txt contains info about the last Applet loaded in IE. All the System.out messages and exception information is stored here when Java Logging is enabled in IE. To enable Java Logging start IE and select View/Options/Advanced.

  Select "Enable Java Logging" check box click OK. Restart IE.
  In NT4 the file in C:\WINNT\Java

- **Question:** Is there a way to reduce the amount of time that it takes to download an applet?

  Answer: There is a way to reduce the amount of time an applet takes to download. What ever classes the Java applet is referring - you cluster them in a JAR file with the help of JAR utility that comes with the JDK version. Check out the help for the options of that utility and make a ".jar" file out of the applets referred classes and images and other relevant data which you want to load.

  Use the archive option of the applet tag and assign the .jar file:

  <applet code="xyz.class" archive="pqr.jar" width=100 height=100>
  </applet>

- **Question:** When I reload my applet my hidden canvas is shown directly! Why?

  Answer: Put mycanvas.setVisible (false); in Start() rather than init()...

- **Question:** I want to be able to print debugging text messages during the whole applet's lifetime. Is there an easy way to do that?

  I'm a beginner in java. Right now i am doing an applet and i want to write messages to the browser window for debugging purposes i.e. to follow how the applet

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

executes. Like when I'm developing a C++ application i usually use lots of "couts" to check values and the programs behavior. Is there an easy way to do things like that when making a Java applet? For me it seems like everything happens in a function called "paint(graphics g)" and that function is only called at the beginning of the applet start. I want to be able to print text messages during the whole applet's lifetime.
Is there an easy way to do that?

Answer: you'd be better off doing a

System.out.println("the value is " + whateverValue);

This will show up in the java console. to see it in ie5, do View->Java Console, and in netscape4.7, do Communicator->Tools->Java Console and it will pop up the java console window.
If you are doing it in applet viewer from dos, it will show up in the dos window you used to call applet viewer.

Ł **Question:** I am writing an applet that will use images. I would like to ship out the images using a jar file that contains all the images that the applet is going to use. I have seen a piece of code that does that in the past, but I don't remember where.

Answer: The following is from:

http://developer.netscape.com/docs/technote/java/getresource/getresource.html

```
import java.applet.*;
import java.awt.*;
import java.io.*;
public class ResourceDemoApplet extends Applet {

  Image m_image;
  public void init() {
     try {
       InputStream in = getClass().getResourceAsStream("my.gif");
       if (in == null) {
         System.err.println("Image not found.");
         return;
       }
       byte[] buffer = new byte[in.available()];
       in.read(buffer);
       m_image = Toolkit.getDefaultToolkit().createImage(buffer);
     } catch (java.io.IOException e) {
         System.err.println("Unable to read image.");
         e.printStackTrace();
     }
  }
  public void paint(Graphics g) {
    if (m_image == null)
    return;
    Dimension d = getSize();
    g.drawImage(m_image, 0, 0, d.width, d.height, Color.white, this);
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

- **Question:** I have made an applet in VJ++ which I have to sign. Is there any tool to do it (both signing and cabbing)..?

  Answer: Signing and archive files are two of the biggest bothers in Java. Everyone uses a different system. A good place to start is:

  http://www.suitable.com/Doc_CodeSigning.shtml

  One of the other bothers is that the unsigned window warning can't be removed by signing an applet for Internet Explorer for Macintosh. And while I am on the subject, the Windows Netscape 4.x system has a bunch of privilege calls:

  http://developer.netscape.com/docs/manuals/signedobj/capsapi.html

  and you need under most circumstances to make Microsoft specific calls too, detailed in links from:

  http://www.microsoft.com/java/security/

  Going through all this will make you want to curse. Unfortunately it is hard to pick a convincing scapegoat. It is true that Microsoft chose an entirely nonstandard CAB system, but it produces archives that are about 40% smaller than JAR files. Signing archive files is a perfect microcosm of the "freedom to innovate" controversy. Microsoft has done a better job but taken away predictability and uniformity. If the Java standards were not controlled entirely by Sun, a Microsoft competitor, perhaps everyone would be using smaller archive files by now.
  --
  Mickey Segal

- **Question:** I want to use more fonts in my applet... say for example Arial... which is not available in the present JDK package...
  How can I deal with it?

  Answer:
  ```
  import java.awt.Toolkit;
  ...
  Toolkit tools = new Toolkit();
  String[] fontList = tools.getFontList();
  ```

- **Question:** How can I slow down my applet?
  I have a game applet that is running too fast on newer systems that have high-end video cards. Its easy enough to slow down the game by having it sleep between thread cycles, but I need to be able to determine how fast a users machine is before I determine how long to sleep for.
  I have been muddling through the documentation but cannot find any calls that will tell my applet what the user's configuration is as regards to CPU speed and other components they may have on their system.

  Answer: Simple create a new Date (), then perform a standard lengthy operation on the order of something that takes about one second on your machine, like a long loop, then create another new Date() and compare it to the first. If it takes 1/2 of the time compared to your machine, then the CPU is probably about 2 times faster. if

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

it takes 3 times the duration compared to your machine, the CPU is probably 1/3 as fast as yours.

Do this dynamically, and it might help with speed changes when there's lots of action happening as well - unless this issue is already being dealt with using threads, that is.
--
by Max Polk

- **Question:** Why do I see applet in applet viewer and do not in a browser? When I try to view my applet on a web page I get the error:

  java.lang.NoSuchMethodError: java/lang/Double: method parseDouble(Ljava/lang/String;) not found

  Which is weird as it compiles fine on Borland and with the JDK using applet viewer Anyone have any ideas what is going wrong?

  Answer: The parseDouble method was only added to Java in JDK 1.2 Browsers typically only support Java 1.1.

  If you have the JRE installed, you can run Java 1.2 applets. But you must also change the HTML code that embeds the applet. Check javasoft.com. I believe they have a program which will automatically change the <APPLET> tag to <EMBED> and add whatever else is needed. It's been a while since I've done applets but I do remember running across a similar problem.

- **Question:** In my applet I have a bunch of gif's in my JAR file. When I try to access a gif using:

  Image img = getImage(getCodeBase(), "image.gif");

  everything works fine under Microsoft Internet Explorer but it does not under Netscape and applet viewer. Of course I do not have any gifs in my CodeBase directory on server.

  Any idea why?

  Answer: Because this is not how you access resources in a Jar file. You need to use getResourceAsStream if you want to access GIFs from Netscape. Look at:

  http://developer.iplanet.com/docs/technote/java/getresource/getresource.html for example code. This same code will work in Sun's Applet viewer.
  --
  David Risner
  http://david.risner.org/

- **Question:** How do I get JVM version in Internet Explorer? When you open the Java Console through internet explorer, it prints the following useful line at the top:

  Microsoft (R) VM for Java, 5.0 Release 5.0.0.3318
  We would like to be able to obtain the above String (or at least the 5.0.0.3318 part of it) through a Java Applet / JavaScript at runtime.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Does anyone know of any handy methods that allow access to this String? I've looked in all the System.properties, but it wasn't there. Is it stored in the user's registry anywhere?

Answer: just for Microsoft's VM!

try :

```
class test{
  public static void main(String[] args){
     String build;
     build=com.ms.util.SystemVersionManager.getVMVersion().getProperty
("BuildIncrement");
     System.out.println("Using build "+build);
  }
}
```

Real Gagnon from Quebec, Canada
* Looking for code code snippets ? Visit Real's How-to
* http://www.rgagnon.com/howto.html

- **Question:** I wonder if there is a way to find out if a button in an applet has been clicked, no matter which of the buttons in an applet it might be.

  Of course I can write, with a particular button (if event.target==button1) but maybe there is a syntax that looks more or less like this (it is an imaginary code just to show what I would like to do)

  (if.event.target.ComponentType==Button) etc.

  I tried a lot of things with getClass but none of them worked

  Answer: Have your applet implement the ActionListener interface, and have every button that's instantiated add the applet as an ActionListener. Then, inside of your applet, have the following method:

```
public void actionPerformed(ActionEvent event) {
  // check to see if the source of the event was a button
  if(event.getSource() instanceof Button) {
     // do whatever it is you want to do with buttons...
  }
}
```

  Darryl L. Pierce Visit http://welcome.to/mcpierce

- **Question:** Could you suggest how to draw one centimeter grid in applet, please? One cm on the screen must be equal to real cm.

  Answer: If you're not all that picky about it, you can always use java.awt.Toolkit's getScreenResolution () to see how far between the lines should be in the grid....that's assuming the applet security allows it.
  But have it _exactly_ one cm, you can't do, since the user can always adjust the display with the monitor controls (making the picture wider/taller/whatever), and no

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

computer that I know of can know those settings.
--
Fredrik Lannergren

Not only that, the OS (and thus Java) does not know if I am using a 21" or a 14" monitor and thus can't know the actual physical size of a given number of pixels. By convention, on Windows monitors are assumed to be either 96dpi or 120dpi (depending on the selection of large or small fonts). Java usually assumes 72dpi. None of these values is likely to be accurate.
--
Mark Thornton

- **Question:** Does anyone know how to or where I can find information about determining if cookies are disabled on a client browser making a request to a Servlet or JSP (or any server side request handler, for that matter)? Also, is there a way to determine whether or not a client's browser has style sheets enabled?

  Answer: To test if the client has cookies enabled, create a cookie, send it, and read it back. If you can't read it back, then the client does not accept them. It's not a clean way of doing it, but it's the only way (that I know if).

  As for CSS, there is no way to know if they allow CSS. Different versions of the browsers support varying levels of CSS. You can get the browser type from the request object and then make decisions based on that.

- **Question:** How can two applets communicate with each other? Have you some examples?

  Answer: You will occasionally need to allow two or more applets on a Web page to communicate with each other. Because the applets all run within the same Java context-that is, they are all in the same virtual machine together-applets can invoke each other's methods. The AppletContext class has methods for locating another applet by name, or retrieving all the applets in the current runtime environment:

```
-----------------------------------------------------------------
import java.applet.*;
import java.awt.*;
import java.util.*;

// This applet demonstrates the use of the getApplets method to
// get an enumeration of the current applets.

public class ListApplets extends Applet {
  public void init() {
    // Get an enumeration all the applets in the runtime environment
    Enumeration e = getAppletContext().getApplets();
    // Create a scrolling list for the applet names
    List appList = new List();
    while (e.hasMoreElements()) {
      // Get the next applet
      Applet app = (Applet) e.nextElement();
      // Store the name of the applet's class in the scrolling list
      appList.addItem(app.getClass().getName());
    }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    add(appList);
  }
}
```

Here are some useful links on applet to applet communication. I don't know if they will solve your problem but these are a variety of good approaches for this type of issue:
http://www.javaworld.com/javaworld/javatips/jw-javatip101.html
http://www.twf.ro/calculatoare/TricksJavaProgramGurus/ch1.htm
http://www.galasoft-lb.ch/myjava/CommTest/backup00/
http://www.rgagnon.com/javadetails/java-0181.html
http://www.2nu.com/Doug/FAQs/InterframeIAC.html

by Mickey Segal

- **Question:** I would like to ask if there's anyway that I can use the same program run as an applet or application...

  Answer: You would have to provide at least a main() for the application part, and init(), start(), stop(), destroy() for the applet part of your program. Your class could simply display the applet within a Frame.

  Example:

```
class Foo extends Frame {
  public Foo(String title){
    //...
    Foo applet = new Foo();
    applet.start();
    add(applet, "Center");
    //...
  }
```

  main()is function of course, not constructor

- **Question:** Is it possible to run a java applet in a dos window (win98 se)?

  Answer: No. A dos window is a character device. You can use the applet viewer program that comes with the JDK though.
  --
  Mike

- **Question:** Is there a simple way to tell if a PC online or not from within an applet?

  Answer: Not without either server-side support or signing the applet, since applets are not allowed to connect to other hosts than the one they are downloaded from. Best approach, I suppose, would be to ping the target from the server.

  However, this is not quite full proof because of "firewalling": my pc, for example, will not answer to pings.
  --
  Michiel

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Question: Is it possible to close browser from applet?

Answer: Yes, use this (tested):

```
//////////////////////////////////////////////////////
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import netscape.javascript.JSObject;
class CloseApplet extends Applet
implements ActionListener{
  protected Button closeButton = null;
  protected JSObject win = null;

  public void init(){
    this.win = JSObject.getWindow(this);
    this.closeButton = new Button("Close Browser Window");
    this.add(this.closeButton);
    this.closeButton.addActionListener(this);
  } // ends init(void)

  public void actionPerformed(ActionEvent ae){
    this.win.eval("self.close();");
  }
} // ends class CloseApplet
//////////////////////////////////////////////////////
```

and the HTML needs to have MAYSCRIPT enabled.

```
//////////////////////////////////////////////////////
<HTML>
<HEAD>
<TITLE>Integre Technical Publishing</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<DIV ALIGN="CENTER">
<APPLET WIDTH="150" HEIGHT="30" CODE="CloseApplet.class"
  CODEBASE="java/" MAYSCRIPT>
</APPLET>
</DIV>
</BODY>
</HTML>
//////////////////////////////////////////////////////
```

Here's the API:

http://home.netscape.com/eng/mozilla/3.0/handbook/plugins/doc/Package-netscape.javascript.html

It's small enough that you could include it in your JAR if you'd like. But most users will even have it on their systems.

It says "Netscape," but I know that IE understands it fine.
--
Greg Faron

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Integre Technical Publishing

- **Question:** Is it possible to run an Applet inside a JAVA application?

  Answer: An applet is just another class that can be instantiated:

  Applet myApplet = new MyApplet();
  where MyApplet is the name of the applet class that you have written and then added to a container of some kind

  myFrame.add(myApplet);

  ..but you need explicitly call the init() method that a browser would normally call "behind the scenes":

  myApplet.init();

- **Question:** I want to change the size(Width x Height) of applet by mouse click on the Button. I used resize(), but it does not get bigger than the area defined by html file. Is there any way to do this?

  Answer: I don't think so. You *could* (perhaps) open your own frame.
  --
  Tim Tyler

- **Question:** I want to bypass the security sandbox so that my Applet loaded from IE can read and write files on the client side:

  - configure the server as trust side,
  - install my own security manager to override those checks.

  Will that work? Do I still need to have my Applet signed?

  Answer: "Core Java II" says on page 716:

  "Once the program installs a security manager, any attempt to install a second security manager only succeeds if the first security manager agrees to be replaced. This is clearly essential; otherwise, a bad applet could install its own security manager.
  Sign the applet and make the Security calls that request the access you need. If the user consents, your code will be allowed. If they don't, it'll fail.
  Be aware that the Netscape security model is not that of the Java specification, and if you want to run there, too, you'll have to find a way of dealing with the discrepancy. Netscape's developer Web site includes the documentation for their model, as well as stub libraries you can compile against."

  For information on the Java security model (Sun sanctioned version), check out these Web pages (chosen from the results of the Google search "Java Security Model"):
  http://java.sun.com/security/
  http://java.sun.com/security/SRM.html
  http://www.sans.org/infosecFAQ/code/java_sec.htm

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I just started to do Java programming. How can I bring the system standard output (what you see on your telnet window) to display in an applet?

  I want to do:

  system(myprogram);
  then I want the screen output to be in my applet.

  Answer: There is no system() method in the standard API. This functionality in Java is provided by the java.lang.Runtime class. Runtime allows you to capture standard output from an external process you started.

- **Question:** How does one remove or replace the message "Java Applet Window" at the bottom of a frame?
  Do not confuse this with the status message at bottom of a browser.

  Answer: This can be done.
  Use the .java.policy file in the user|home directory with following contents:

  ```
  grant {
      permission java.awt.AWTPermission "showWindowWithoutWarningBanner";
  };
  ```

  --
  Ganesh

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

3. Databases & Beans

- **Question:** Anybody does know a freeware JDBC driver for a DSN-less connection to MS SQL-Server? Would even consider a "cheapware" version.

  Answer: Go to http://industry.java.sun.com/products/jdbc/drivers and search for Microsoft SQL Server. Any Type4 (i.e. pure Java) driver should work without a DSN.

  The only free one I'm aware of is at http://www.freetds.org - but it is rather limited in what it can do. You'd need to try it out to see whether it fits your requirements.
  --
  Stefan
  P.S. DSN - Data Source Name

- **Question:** How many kinds of enterprise beans do Java has?

  Answer: Java (J2EE) has three kinds of Java beans:

  1. Session beans -represent transient conversations with clients. The data will be gone after a session is over
  2. Entity beans -represent persistent data and let to store data in a database.
  3. Message driven beans - combining features of a session bean and Java Message Service (JMS). They provide possibility to receive asynchronous JMS messages

- **Question:** Any idea how to lock a Session ID? Ho to prevent its usage again?

  Answer: You do need to trace session IDs to avoid theirs repetition if you use our approach to ID constructing from the tip above. Your IDs always will be unique, since some of data never repeats (such as Date)

- **Question:** I am reading my values from a db which is supposed to be a string, but sometime they can be doubles or something else...
  How could I check if a string was double?

  Answer: You can try to try to parse the String into a double and catch the exception:

```
String mystr="1.6";
try {
   Double dbl = new Double(s);
   // do smth. here
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
catch (NumberFormatException ex) {
  // d smth. else here
}
```

- **Question:** I have developed my program with MySQL and Java. It runs locally. Now I want to make it an online application, including MySQL. But how do I get connection to database?

Answer: No problem! Your database always runs "online". Even when it was running locally you connected to it by using "localhost". Now you need to use url.

Something like this:

```
DriverManager.getConnection(url, user, password);
```

Question: My old program uses beans. When I recompiled it with Java 1.4 I found that my custom attributes in BeanInfo sometime disappears. I am sure that they were set to some my values but after while they are gone and I get null...

Answer: Since Java 1.4 a BeanInfo can be garbage collected when no direct references to it exist and the system is low on memory. in your case probably BeanInfo got collected and getBeanInfo method returns now custom attributes. Java doc for release 1.4 recommends next:

> To avoid this problem, if you store custom attributes inside a BeanInfo you must ensure that the BeanInfo is correctly initialized with those attributes every time you retrieve the BeanInfo. The following is an example of initializing a BeanInfo class with a custom property in the bean descriptor. The code would be similar for a custom attribute in a property descriptor, method descriptor, or event set descriptor.

```
BeanInfo beanInfo = getBeanInfo(SomeBean.class);
BeanDescriptor beanDescriptor = beanInfo.getBeanDescriptor();

/*
 * Before using the BeanInfo, check to see if our custom
 * property has been initialized.  (Even if we initialized
 * it before, if the BeanInfo has been garbage collected,
 * then we need to initialize it again.)  Since our custom
 * property's value could be null, we define another property
 * to tell us if the custom property is initialized.
 */

if (beanDescriptor.getValue("myProperty_init") == null) {
   beanDescriptor.setValue("myProperty", someValue);
   beanDescriptor.setValue("myProperty_init", Boolean.TRUE);
}
```

- **Question:** I have a binary flat-file database that I need to access via JDBC. How do I create a driver that I can use?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: Read SUN's specifications on

http://java.sun.com/products/jdbc/

As written there "the new JDBC API, it also provides access to other tabular data sources, such as spreadsheets or flat files..."

Also take a look at RmiJdbc on:

http://www.objectweb.org/rmijdbc/index.html - open source software.

- **Question:** Why do we need to use Class.forName() to load a JDBC driver into the memory?
I did not understand idea behind of this...

Answer: Class.forName("MyClass") is used for loading the classes dynamically. JVM executes all its static blocks after class loading.
Everything you need to be initialized put into static block..
All JDBC Drivers have a static block that registers itself with DriverManager and DriverManager has static methods only.

- **Question:** I just want to know which programs and virtual machines you have to have to make and run enterprise java beans...

Answer: To compile and run Enterprise JavaBeans, you need a couple of things. First, you need the J2EE SDK. This kit includes APIs full of packages which are considered extensions to the standard Java language APIs, as well as other tools, which come with the J2SE SDK, which you should already have. Install the SDK and make sure its jar file is in your development environment's classpath.

Second, you need a container, which in this case you can also refer to as an application server, though technically a container is just one part of the server. The container acts as a liaison between the client object and the Enterprise JavaBean. When you talk to an Enterprise JavaBean, you actually talk to a proxy (a substitute), and the proxy, which knows how to do networking stuff, talks to the container, which in turn talks to the actual implementation object which is what you think of when you think of an Enterprise JavaBean.

The J2EE SDK, fortunately, comes with a server/container, as well as a GUI-based tool which allows you to deploy your Enterprise JavaBeans in the server. See http://java.sun.com/j2ee .

Third, you need a lot of patience. The learning curve is rather steep unless you have a lot of experience doing network programming. Enterprise JavaBeans are designed to abstract out networking and storage logic, which ends up being very helpful, but is confusing at first, because so much happens behind the scenes that is not explicitly controlled by your code. For example, when you deal with a single Enterprise JavaBean, at least five different objects are actually being instantiated!

But it's great once you get past the initial learning stage, which can last a while. There are lots of good books on EJB, but I found Ed Roman's "Mastering Enterprise JavaBeans" to be a great primer.
--

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Erik

- **Question:** I am reading my values from a db which is supposed to be a string, but sometime they can be doubles or

  something else...

  How could I check if a string was double?

  Answer: You can try to try to parse the String into a double and catch the exception:

  String mystr="1.6";

- ```
  try {
     Double dbl = new Double(s);
     // do smth. Here
  }
  catch (NumberFormatException ex) {
     // d smth. else here
  }
  ```

- **Question:** I'm having a hard time figuring out what are the differences between Enterprise Java Beans and Java Beans.
  Is there a definitive difference between the two?

  Answer: Definitely. JavaBeans are really nothing more than classes that have no-args constructors and follow certain naming conventions (and/or provide a BeanInfo class) to identify properties, methods, and events. JavaBeans were designed for plugging into GUI design tools. Technically speaking, lots of things are JavaBeans... though whether they are intended to be used that way is another matter altogether.

  Enterprise JavaBeans are not really used as JavaBeans at all. They run on a different server, in a special EJB container that provides a bunch of restrictions on their class hierarchy, their fields and object relationships (particularly if CMP is used), the Java language features that can be used if you write them, etc.

  They are, of course, far from a GUI design thing. They are also, IMHO, far from a good idea and I'd avoid them if at all possible.

  --> to be continued tomorrow (end of Part 1)

- Part 2.

  > Please explain why to avoid them in more depth...
  Basically, the problem is that Sun has started reflection "activity". I use that term to describe the unnecessary use of reflection when a good solution with static typing would be preferable.

  Examples: Standard RMI has always used interfaces to act as common types between client and server. EJBs abandon this, and just say "well, define all the same methods" with no static checking that you've done so. You then have to write a bunch of "ejbCreate" methods, with exactly the same signature as "create" methods,

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

  ```

but needlessly renamed. (Well, you kinda would have to name them in order to stuff them into a class where they don't belong, which is exactly what EJB does.) Then EJBs examine my object's fields (which are supposed to be private), and sometimes the whole thing stops working because I have a field of a type that the container doesn't like.

The result is something that shares basic syntax with Java, but which is really a different beast altogether, with no well-organized bit of documentation on use (because everything is dynamic through reflection instead of working with well-defined methods in well-defined classes for which documentation can be written), and strange and unreasonable constraints on coding.

That said, EJB containers are the only environments that provide anything like CMP, for example. I'm working on a better-designed replacement for EJBs' CMP that doesn't rely quite so heavily on reflection, but I'm unaware of anything widely available.
--
Chris Smith

--> to be continued tomorrow (end of Part 2)

- Part 3.
It was a poor naming decision on Sun's part. There is no similarity between Java Beans and Enterprise Java Beans except for their name.

All they managed to do was cause confusion just because they liked the cutesy name.

The confusion is reduced when you realize that "Bean" in Java is the cutesy name for that software engineering term, a.k.a. buzz word "component" that is something intended to be re-usable as is without change, the software equivalent of the hardware engineers chip.

One note: a JavaBean is intended for use in some type of building tool, but a bean or a set of beans may not have anything to do with the GUI of the resulting application. Most often the intent is that the bean is _configurable_ using some GUI. The typical examples of non-graphical beans are Database access JavaBeans which result in nothing in a GUI, but may communicate with other Beans which are part of the application GUI.

I was annoyed a few months back when I attended a Sun training, which included coffee bean looking icons, plus various other curious bits of graphics which to me just added clutter. I still don't know what some of the silly figures where supposed to be :-). Some simple UML (some boxes and arrows) would have been much clearer.
--
Comments by Tom Almy, Paul Hill

- **Question:** XBeans?

**Answer:** XBeans are Java Beans. An Xbean is a software component that takes XML as input, processes it in some fashion and then passes XML on to the next Xbean.
---
-----> \ XBean \ ----->
XML --------- XML

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Java Bean technology supports the packaging, reuse, connection and customization of Java code. With the appropriate set of XBeans and a Java Bean design tool, it is possible to build useful distributed applications with little or no programming! Please read the white paper "Creating Distributed Applications Using XBeans" for a more detailed discussion of XBeans and applications that use them.
http://www.xbeans.org/ Xbeans.org is an open source project.

All used above!

- **Question:** I want to use MS Access databases in my java app. Where do I start and where can I get the drivers, I am fluent in SQL with Active Server Page experience, so that isn't a program, its just setting up the DB connection in Java.

**Answer:**

1. From the ODBC control panel, define a system data source name for the specific MS Access database you want to use. Make sure it's a system DSN, not a user DSN. Assume you name it "foo".

2. In your Java application, load the JDBC driver:

```
try {
   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
catch (ClassNotFoundException e) {
     // Shouldn't happen - it comes with the JRE
}
```

3. Create the DB connection using the name of your system DSN from step 1:

```
Connection con = DriverManager.getConnection("jdbc:odbc:foo");
```

That's it. From there on, the connection behaves like a JDBC connection to any other kind of database. See the tutorial for details.

Remember to close the connection - it's best to enclose step 3 in a try/finally block:

```
Connection con = null;

try {
   con = DriverManager.getConnection("jdbc:odbc:foo");
   // Do your stuff
}
finally {
    if (con != null)
    con.close();
}
```

--
Phil Hanna
Author of JSP: The Complete Reference
http://www.philhanna.com

- **Question:** I don't think my brain just isn't getting the whole picture...
If I write a Java app that is going to be talking to Oracle, is the only method of getting the Oracle server to do anything via SQL calls?

Assuming that my Java App is on a laptop, and talking (I guess TCP/IP) to the Oracle

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Server, what processing/data access options do I have?

The Oracle site seems to talk about a Java API to Oracle, but inside how would my laptop app be getting the data? Is this API just a wrapper around SQL calls? Is there an "Included into my Laptop App" something that will allow me to create objects based upon rows of data stored at the server, or do I have to get the data, and then construct the Java objects myself?

I guess my simple question is, "Is SQL the only data access method to Oracle that I've got?"

If there are any successes I would like to hear them (not from Oracle though, I have a hard time buying people blowing their own horns).

Answer: If you want to know how to get >data< interaction between an Oracle database (or any other 'industrial' database) and your Java app running on a laptop, the primary & recommended way is to use a SQL API. The possibilities include JDBC and SQLJ. Both are industry standard APIs and are explained at http://technet.oracle.com/tech/java/sqlj_jdbc

Other methods for interacting for interacting with the JVM inside the database include XML, JSP, CORBA, EJB, etc. But most fall back on JDBC and/or SQLJ for actual interaction with database objects. Recommended page to start looking at this is

http://technet.oracle.com/tech/content.html

Oracle does provide a few things to help you out including (but not limited to):

JDeveloper is their Java IDE and it has a number of enhancements & wizards to try to make life a lot simpler for various types of operations;

and BC4J (Business Components for Java) provides a framework for creating and using 'business components' that require database interactions.

Oracle database built-in capabilities also minimize or eliminate special custom Java code; intermedia is included in the database and handles virtually all multimedia interactions as well as text indexing and text searches a lot easier than custom Java code;

spatial option simplifies interaction with polygon and spatial coordinate manipulation, management and searching;

iFS does a fair bit of standard document management while making the database look like a NFS disk (or network drive in Microsoft environment).

However, most of these use some form of SQL API (again JDBC or SQLJ) as the Java interface.

And since the Oracle9i Application Server includes a report and ad-hoc query capability, that can eliminate or minimize Java for information distribution.
--
Hans Forbrich

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I used JDBC driver to connect Microsoft SQL server Database. It is no problem besides display Chinese word...

  Answer: You can try

```
public String fromDB(String in) {
    try{
        return new String(in.getBytes ("Iso8859-1"),"Big5");
    }
    catch(Exception e){
        return "";
    }
}
```

  This is the problem at the encoding of string.

4. Distributed systems

- **Question:** Has anyone ever tried anything like this or am I asking for trouble trying to write a program like this?

  I plan to use JBuilder to create a Java GUI that will use Perl to invoke system calls.

  The GUI will be run in Windows (NT) while the system calls will be invoked in Unix.

  Answer: Sure, why not? Seems to me it should be quite doable. Use Java code to build the GUI and cross the network (for instance using RMI), then invoke the Perl interpreter as an external process, or possibly use JPerl (see http://www.perl.com/CPAN-local/authors/id/S/SB/SBALA/) from there. Or use different distributed objects architecture to connect Java and Perl objects over the network...

- About serialization...

  If I have a class that implements the Serializable interface, but it has member variables which reference objects that do not implement the Serializable interface, it appears that I can't serialize an instance of the class. I keep getting:

  java.io.NotSerializableException

  for one of the objects referenced by a member variable.

  Am I correct, or am I just missing something. Also, if anyone knows a work-around to serialize non-serializable objects, I'd like to hear about it. Unfortunately, I have no control over the classes I'm trying to serialize, so I tried putting a serializable wrapper around them, but that didn't work.

  Answer: Do you really need to serialize those members of your class which aren't serializable? In other words, make them private:

```
class Foo implements Serializable {
   private Bar bar;
}
```

  Do you *need* to maintain the state of the 'bar' variable when

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

serializing/deserializing Foo? If not, simply declare 'bar' as 'transient' and it will be ignored during serialization.

- RMI versus Socket communication...
I wish to get Java talking to C++ across a network.
Does anyone have any thoughts in terms of performance, ease of development etc.

  Wrapping the C++ side with JNI and using RMI for the communications.

  versus

  Writing sockets code and communicating via http?

  Answer: It depends of what kind of application you're writing but I think about the following:

  - with RMI you can have remote REFERENCE instead of having to transfer the entire object through the network. The object has just to implement Remote. So it spare bandwidth and is good for performance. This is impossible to do if you do through a socket connection, you've to send the all object.

  - You've not to take in charge the serialization (which could be not so easy depending of your object structure), neither the connections, etc... All of that is taken in charge by RMI.
  - the performance are GOOD (even a bit more than that) three good points to use RMI, isn't it?

  The difficulty added by RMI is the configuration of both client and server (distribution of stubs, rmiregistry, what's happen if firewall). Depending of the environment all of that can be either easy or complicate.

  But once that all of that is in place you can extend your application easily, so it's much more flexible and scalable.

  If your need are small perhaps that you could do your own connection system (but for me it's less scalable and more bandwidth consuming and so less performance).
  --
  Francois Malgreve

- **Answer 2**: I have done both. If your communication scenarios are diverse and could keep changing, using a remote technology like RMI can help. If the operations are few and/or not likely to change you can save the JNI complexity. Not that it is really hard it just can be fun keeping the JNI code in sync with the C++ code.
  --
  Bret Hansen

- **Question:** I need to communicate some data (string) from a Java Applet to another ASP page in the same frameset. I would like to avoid a server roundtrip and do it all with JavaScript if possible.

  Therefore I would like to call some JavaScript from a Java Applet. It looks like it is not possible without a Netscape package. Is that true? Is there a simple

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

implementation of the same functionality (source code) which I could incorporate in my applet?

Answer: Those Netscape packages are part of the current VM of both Microsoft IE 4+ and Netscape 4+. So, by adding the MAYSCRIPT tag to your Applet declaration, in the Java code you can obtain a handle to the document and call functions in it.
--
by Tom Hall

- **Question:** I'm researching methods by which one JVM can interact with another JVM, which is running on the same machine.

  I know that there are various network models, which can be applied if a JVM needs to talk to another one across a network, but in addition to these (which could I guess be applied to JVMs on the same machine) I wondered if you knew of a system of JVM communication that requires less system resources, where the JVMs are both running on the same system.

  Answer: CORBA, RMI, HTTP, sockets....
  But if you have no TCP/IP stack on your platform, so for Windows it could be clipboard...
  --
  by Dmitry

- **Question:** I have a question about sending a reference to the object via the socket...
  I have a question about sending a reference to the object via the socket. Two threads are communicating via sockets running on the same machine. I don't need to send the whole object, but I need to send just a reference.

  Does anyone know how to do that?

  Answer: Reference to an Object? A reference is only valid within the same memory space! If you want to be able to invoke methods on an object remotely, then you will need to use a remote technology like RMI, CORBA, or some such.
  --
  by Bret Hansen

5. File Systems

- **Question:** How does Java read the text files? Can Java read the files that are in other formats? Is the read file method in Java only recognizes the file in .txt or other text format?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Java can read any text file (using a java.io.FileReader for example), the attribute at the end is an indictor and thus is not relevant as long as the actual code read is in the correct format. I It can read files that are in other formats bytes etc and if you have a weird format you could extend the IO mechanism with some work to work with that.

- **Question:** What's the preferred way to copy files in Java?

Renaming is easy, since java.io.File provides a method for that.
But I didn't find a method for copying.

Answer:

```
public static void copyFile(File srcFile, File dstFile)
      throws IOException {
   BufferedInputStream in =
                    new BufferedInputStream(new FileInputStream(srcFile));
   BufferedOutputStream out =
                     new BufferedOutputStream(new FileOutputStream(dstFile));
  byte buffer[] = new byte[BUFFER_SIZE];
   int count;

   while ((count = in.read(buffer)) != -1) {
     out.write(buffer, 0, count);
   }

   in.close();
  out.close();
}
```

Note that this doesn't close either of the streams if an exception is thrown - which is precisely the kind of thing that finally blocks are for. (Be careful to make sure you close both streams even if one of the calls to close throws an exception though and not to try to call close() on null if the streams haven't been set up...).

It's not a full-fledged solution. My proposal is only a draft.
--
Peter

- **Question:** I create two file objects and use "renameTo()" method to rename second file to first one. Will it change the content and object reference?

Answer: No.

```
File f1 = new File("aaa");
File f2 = new File("bbb");
f1.renameTo(f2);
```

It changes the name but not object reference

f1 file will be renamed to "bbb", but if you use "getPath()" method you will see that the path is still as it was "aaa" file.

- **Question:** Is it possible to choose a directory path instead of a file path in a Swing FileChooser dialog box?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Assume you have instance chooser of type JFileChooser, invoke...
chooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);

- **Question:** Is there anyway to find out the creation date of an existing file?

  Answer: The only thing you can get with pure Java is the last-modified date, which may or may not be the same as creation date.

  Probably the reason the API is limited in this fashion is because not all file systems even store the creation date.

  Windows does store it, but you'll need JNI (or possibly some horribly ugly hack with Runtime.exec() and the DOS "dir" command) to get at it.

- **Question:** Does anyone know the way to retrieve the path from where has been loaded the main class of an application, not the path from where has been launched the virtual machine?

  Answer: Please use something like this:

  path = System.getProperty("user.dir");

- **Question:** How to erase the content of (text) file without create/open file again (only do "new FileOutputStream(...)" once)?

  Answer: Try java.io.RandomAccessFile.setLength (0) if you're using JDK 1.2 or higher. If you don't have to keep the file, it may be easier to use:

  java.io.File.delete().

- **Question:** When I copy an MS Excel file to another Excel file, I'm getting garbage characters together with the text.

  How can I avoid it and copy only the real content?

  Answer: Don't use a FileReader and characters to do the copying. An Excel file contains control characters that cannot be interpreted as text. Use a FileInputStream and bytes, for example:

```
public void fCopy(String inputFile, String outputFile) {

  try {
    FileInputStream in=new FileInputStream(inputFile);
    BufferedInputStream reader=new BufferedInputStream(in,4096);
    FileOutputStream out=new FileOutputStream(outputFile);
    BufferedOutputStream writer=new BufferedOutputStream(out,4096);
    byte[] buf=new byte[4096];
    int byteRead;
    while ((byteRead=reader.read(buf,0,4096))>=0) {
      writer.write(buf,0,byteRead);
    }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
        reader.close();
        writer.flush();
        writer.close();
      }
      catch (Throwable exception) {
        exception.printStackTrace();
      }
    }
```

- **Question:** My application tries to read html files from web server. Some of files are protected by an authorization procedure - server asks for username and password.

  What could I do?

  Answer: Connecting via a socket to port 80 of the web server, put your necessary info into HTTP message.

  You need to provide an authorization field in the header section of the HTTP message, with the user and password. They are encoded using base64 format. I do not remember details of protocol... You can read more in RFC that describes the HTTP protocol.

- **Question:** FilePermission cannot Handle Symbolic Links? I was trying to change file permissions on a link but it did not changed! I am owner of a link...

  Answer: Although you can fully own a soft link, the permissions that apply for any action are the ones of the target. OS prohibits a user from changing the permissions of the files being linked to. Otherwise it could be easy to break a security: just link to any file and change permission :-)

- **Question:** When creating a new file, is it possible to control whether or not an existing file with the same name is/is not overwritten? I haven't been able to answer this by looking at the java.io API.

  Answer: There is a method in File to atomically create a new file which will fail if the file exists... You can use this to try creating the file, and if it exists already don't ever open the FileOutputStream to write contents.

- **Question:** Are existing “Compound Files” in Java?

  Microsoft has made an API where you can have a whole "virtual" file system inside a single file on the real file system. I think they call it Compound Files - I have also heard about the concept under the name of Structured Storage.

  Do any of you know if something like that exists in a Java-library? I have to work for all platforms! (It can of cause be implemented with the use of Compound Files in its Windows implementation)

  Answer: jar files are essentially the same. Have a look at java.util.jar.JarFile

--

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

by Bret Hansen

- **Question:** Is it possible to redirect the System.out.println to a file?

  Answer: Connect a PrintStream to the file, and then call System.setOut (PrintStream out) that reassigns the "standard" output stream.

- **Question:** Could some kind person please tell me how to save the object as a file so as the same program can load it again?

  Answer: try this program. It saves object into file:

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;
import java.io.IOException;

public class Save{
  public void saveMyObject(String filename, Object obj) {
    File myFile = new File(filename);
    try {
      FileOutputStream fileOutStr = new FileOutputStream(myFile);
      ObjectOutputStream outStr = new ObjectOutputStream(fileOutStr);
      outStr.writeObject(obj);
      outStr.close();
    }catch (IOException e){
      System.out.println("?!!!!!!");
    }
  }
  public static void main (String args[]) {
     Save s = new Save();
     Object myObject = new Object();
     String test = "test";
     myObject = (Object)test;
     s.saveMyObject("myfile", myObject);
  }
}
```

  If you open myfile you will see that this object includes our string "test".
  In the same manner you can read this object from file...

- **Question:** Can anyone write me a short method that lets me know what files are in a particular directory? For example, I want to know that directory, d:/temp/aaa, has files a.txt, b.java, b.class. Also related to this, how do I find out what folders I have?

  Answer: use our program as a base and add checking for the files and directories you need to find!

  here it is:

```
import java.io.File;
public class Save{
  public void showDirectoryList() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    File dir = new File("d:/temp/aaa");
    File[] list = dir.listFiles();
    for (int i=0; i<list.length; i++) {
      if (list[i].isFile()) {
        System.out.println("File "+list[i].getName());
      } else if (list[i].isDirectory()) {
          System.out.println("Directory "+list[i].getName());
      }
    }
  }
  public static void main (String args[]) {
     Save s = new Save();
     s.showDirectoryList();
  }
}
```

- **Question:** How do I delete a file in Java? I have programmed a Java application that needs to delete a file, but I couldn't find anything on the topic.

  Answer: in java.io.File there is delete() method:

  public boolean delete()

  It deletes the file or directory denoted by this abstract pathname. If this pathname denotes a directory, then the directory must be empty in order to be deleted. Create a new File object representing the file, and then use the delete () method. If you use the Microsoft JVM, make sure the file is not shared before you delete it, otherwise that will not work (the delete () method returns "false").

- **Question:** I need to read C:\test.doc for example and store it somewhere for my future use. I don't know anything about security. Part 1

  Answer: If you want to read a file on the local file system in an applet, you are going to digitally sign the applet, and the user of the applet is going to indicate trust in the signature. How you go about this depends on a number of questions, like:

  In which browser(s) will the users be running the applet? Will a Java plug-in be installed in the browser? Can the user install a plug-in?

  Here are a few tips on some of the sticky points of signing applets...

  If you are using the jar signing tools that come with the JDK, you'll find that they will only work with the Java plug-in. Netscape and Microsoft IE don't salute the Sun way of signing applets.

  If you can rely on your users to install the Java plug in, you can sign applets using the JDK 1.1.X javakey command. You will also have to have them import your certificate into their 'identitydb.obj' file in their home directory, and mark it as trusted. If importing the certificate is too much to ask, and it usually is, you could create an 'identitydb.obj' containing the certificate, already marked as trusted, and have them download this to their home directory. Be careful not to distribute the 'identitydb.obj' that you used to sign the jar, since that would allow anyone to sign code as you!

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

JDK1.2.X works in a similar way, but the 'javakey' command has been replaced by 'jarsigner' and 'keytool'. In addition, 1.2 adds granularity to the security model, so that you can assign privileges outside of the sandbox via a policy file. For example you could allow an applet to read files, but not write them. JDK 1.1.X security is kind of an all or nothing proposition.

If you can't count on the Plugin being there, but you know your users will be using Netscape, you can sign use the Netscape signtool. Netscape's signtool is a little different than Sun's. With Netscape's, you put all of your class files into a directory, like signdir, and then run something like this:

./signtool -k MyCert -Z MyApplet.jar signdir

The 'MyApplet.jar' file will be created, and, of course, you would already have to have MyCert in your security database. You can generate a test certificate through signtool, but it's only good for a couple of months.
--
Joe Morrogh - Excite@Home - The Leader in Broadband http://home.com/faster

Ł **Question:** I need to read C:\test.doc for example and store it somewhere for my future use. I don't know anything about security. Part 2, Part 1 was published yesterday...

Answer 2: I think they want you to go out and buy a cert from Verisign or Thawte, etc. You can also generate your own signing certificate through openSSL, or some other package, but, if you do, you'll need to have them import it into their database as a trusted signer. This can be done with a simple Perl script which sends the certificate with a header of:

"Content-Type: application/x-x509-ca-cert"

You must also add some code to your applet to use the Netscape way of signing. You can download the file 'capsapi_classes.zip' from Netscape. This file contains the Netscape Capabilities API. Basically, all you need to do is import the package and enable privileges as you need them. It is important to enable privileges in the same method (i.e. same frame stack) in which you are going to use them. You cannot just enable all the privileges you need in the init method of your applet. The code looks something like this:

```
import netscape.security.*;

public void MyMethod() {
  try {
    PrivilegeManager.enablePrivilege("UniversalFileRead");
    FileInputStream fis = new FileInputStream("myfile");
  } catch (Exception e) {
    System.out.println(e);
  }
} // end MyMethod
```

You'll need to add 'capsapi_classes.zip' to the archive attribute of your applet tag. I also believe it is possible to grant privileges to a codebase in Netscape without signing. This is done by editing the 'prefs.js' file on the client. Obviously, this is not

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

possible on the internet, but, for an intranet environment, this could be a solution. Anyway, when an applet tries to enable a privilege, the Netscape browser will pop up a window asking the user if they wish to grant the privilege. This decision can be persistent.

The user can manage the privileges they have granted by clicking on the padlock icon in the bottom left corner of their browser window, and selecting the "Java/JavaScript" section.

If your users may be using Microsoft IE as well as Netscape, you'll have to learn the Microsoft security API. I haven't done this yet, but I believe it is possible to code an applet so that it will work in either browser.

Another alternative is to code different versions of the applet, and build the page containing the applet dynamically, based on the "User-Agent". I have found that it is possible in IE to grant universal privileges to an applet, even if it is not signed, but I sure wouldn't recommend doing that.
--
Joe Morrogh - Excite@Home - The Leader in Broadband http://home.com/faster

- **Question:** Is there any method like writeLine ()?

  Answer: For writing text to files, you can use a FileWriter; for binary data, use a FileOutputStream. Both are in the java.io package. Have a look in the docs, all you need is there. Oh, you might want to wrap your writer/outputstream in a buffered one for more efficiency. Something like this:

```
FileWriter fw = new FileWriter("D:\temp\test.txt");
BufferedWriter bw = new BufferedWriter(fw);
bw.write("Some text on the first line");
bw.newLine(); // have the Writer insert the newline
bw.write("Some more text, followed by an explicit newline\n");
bw.close();
```

  Note, that while \n will probably work, newLine() inserts the platform's own line separator (i.e., \r on Mac, \r\n on Windows, \n on *nix).
  --
  Michiel

- **Question:** I want to copy a set of gif files from one directory to another. If I use any of the File Reader/Write classes of java.io.package, gif files are being copied but are distorted. Most probably because Java reads in character mode. So, how can I copy gifs?

  Answer: use FileInputStream FileOutputStream instead.

- **Question:** Is there a newsgroup on how to use Java for CDs? I mean not CD that install something but CDs that you browse.

  Answer: Strictly speaking, there is no difference between "CDs that install something" and "CDs that you browse". All CDs contain the same file structure. It's up to the operating system to decide what it does with those files. Windows includes functionality to start a special file on the CD automatically, typically to install

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

something.

You read them just as you would any other disk files. To create them, usually you provide a list of files to a creator program.
--
Joona Palaste

Ł

6. Graphics, AWT, Swing

Question: Just now I found that Java for some Swing components support just "Drag" functionality, not “Drop”. Sounds very funny :-) What can I do with "Drag" if I cannot "Drop" it somewhere?

Answer: You are right:
Quick Reference: Components That Support DnD

Component	Supports Drag	Supports Drop
JColorChooser	X	X
JEditorPane	X	X
JFileChooser	X	
JFormattedTextField	X	X
JList	X	
JPasswordField		X
JTable	X	
JTextArea	X	X
JTextField	X	X
JTextPane	X	X
JTree	X	

While drop is not supported it gives some useful info to user, for example in JFileChooser as the mouse is moved over the list, the potential drop point is indicated by highlighting the file/directory under the cursor.

To fully support imports of data the developer must write and install a custom TransferHandler.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** Could you tell me what is advantage of Spinner over Combo box?

  Answer: Really no advantages, excluding one: spinners are sometimes preferred because they don't require a drop down list that can obscure important data. Indeed, in Spinner component you can "spin" thousands of values (for example year). Let's now imagine drop down window which contains all years of human history :-)

- **Question:** the class Robot was introduced in Java version 1.3 and belongs to awt (java.awt.Robot). Why? I got an impression that all new GUI features come to Swing and Robot class is GUI related class...

  Answer: The reason is simple: The Swing toolkit is a fully-featured UI component library implemented entirely in the Java programming language. The Robot class is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed. The primary purpose of Robot is to facilitate automated testing of Java platform implementations.

  Using the class to generate input events differs from posting events to the AWT event queue or AWT components in that the events are generated in the platform's native input queue. For example, Robot.mouseMove will actually move the mouse cursor instead of just generating mouse move events.

- **Question:** How can I colour in real time user text input?

  Answer: Sometime it is easier to use already developed libraries and save money and time.
  An good example of such method is JSyntaxColor.
  JSyntaxColor is a library for colouring in real time user text input.

  Main features:

  Standard EditorKit
  Simple property file for each syntax description
  Real time switch between syntax descriptions
  Very compact and fast
  Inner cache for limiting parsing scope

  Tokens libraries customizable by user
  Color, Underline, Border attributes support
  Sample for Java/SQL/Formula syntax

  Many samples for API interesting parts
  Open source for the registered version
  JDK 1.2 compliant

  Evaluation copy can be found at: http://www.japisoft.com/syntaxcolor/index.html

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I want to get mouse move and click events when the mouse is outside of the Java application. The standard mouse adapters and listeners only get such events when the mouse cursor is over a Java application, not outside it. Is it possible?

  Answer: No, it is not possible... In order for a MouseListener to know when the mouse exited a component, it first needs to know the position of the application's GUI on the screen and also position of the cursor on the screen. Yet SUN provides no methods to access these coordinates...

- **Question:** I design User Interface for server applications. Are there any libraries, APIs, rules should I use/follow to make the development easier and standardized?

  Answer: Yes, the JavaServer Faces Technology!
  JavaServer Faces technology simplifies building user interfaces for JavaServer applications. With the well-defined programming model that JavaServer Faces provides, developers of various skill levels can quickly and easily build web applications by: assembling reusable UI components in a page, connecting these components to an application data source, and wiring client-generated events to server-side event handlers. With the power of JavaServer Faces technology, these web applications handle all of the complexity of managing the user interface on the server, allowing the application developer to focus on application code.

  JavaServer Faces technology includes:

  A set of APIs for: representing UI components and managing their state, handling events and input validation, defining page navigation, and supporting internationalization and accessibility.

  A JavaServer Pages (JSP) custom tag library for expressing a JavaServer Faces interface within a JSP page.

  Designed to be flexible, JavaServer Faces technology leverages existing, standard UI and web-tier concepts without limiting developers to a particular mark-up language, protocol, or client device. The UI component classes included with JavaServer Faces technology encapsulate the component functionality, not the client-specific presentation, thus enabling JavaServer Faces UI components to be rendered to various client devices. By combining the UI component functionality with custom renderers, which define rendering attributes for a specific UI component, developers can construct custom tags targeted to a particular client device. As a convenience, JavaServer Faces technology provides a custom renderer and a JSP custom tag library for rendering to an HTML client, allowing developers of Java 2 Platform, Enterprise Edition (J2EE) applications to use JavaServer Faces technology in their applications.

  Ease-of-use being the primary goal, the JavaServer Faces architecture clearly defines a separation between application logic and presentation while making it easy to connect the presentation layer to the application code. This design enables each member of a web application development team to focus on their piece of the development process, and provides a simple programming model to link the pieces together. For example, web page developers with no programming expertise can use

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

JavaServer Faces UI component tags to link to application code from within a web page without writing any scripts.

This and more description is published on:
http://java.sun.com/j2ee/javaserverfaces/

- **Question:** What is Auxiliary Look and Feel?

  Answer: An auxiliary look and feel is like any other look and feel, except that it doesn't have to provide the complete support that a default look and feel must. For example, an auxiliary look and feel that supports just text-to-speech feedback doesn't need to provide any code for painting. Also, it might not need to support all components -- JSeparators, for example, might be ignored.

  Auxiliary look and feels tend to be simple, so developing one can be easier than developing a visual look and feel. The developer can concentrate solely on providing the specialized functionality.

  Because the primary purpose of an auxiliary look and feel is to enhance the default look and feel, auxiliary look and feels tend be nonvisual. Since an auxiliary look and feel is a genuine look and feel, however, there is nothing to prevent it from rendering information on the display.

  Just like for any other look and feel, you implement an auxiliary look and feel by writing a subclass of javax.swing.LookAndFeel and creating subclasses of the FooUI classes defined in the javax.swing.plaf package.

  used info from "Using the Multiplexing Look and Feel " -
  http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/plaf/multi/doc-files/multi_tsc.html

- **Question:** What are some general recommendations for developing auxiliary look and feels?

  Answer:

  Use the installUI method to perform all initialization, and the uninstallUI method to perform all cleanups.
  Don't extend visual look and feels.
  Override all UI-specific methods your UI classes inherit.
  Details on recommendations are here:
  http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/plaf/multi/doc-files/multi_tsc.html

- **Question:** What can I do if I want to use User Interface (UI) Objects from different look and feels simultaneously in one UI?

  Answer: The classes in the javax.swing.plaf.multi package implement a multiplexing look and feel. A multiplexing look and feel transparently creates -- and simultaneously supports -- UI objects from several different look and feels in

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

response to a component requesting its UI object (with the getUI method).
About what you gain read more here:
http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/plaf/multi/doc-files/multi_tsc.html#overview

- **Question:** Was Swing performance improved in Java 1.4.2?

Answer: Start-up time is faster now ~by 20%. Also JFileChooser had a bug before and now it performs 300 times faster!!!
It could be good if all Swing classes have got such improvement :-)

Release 1.4.2 introduces support for the standard Microsoft Windows XP appearance as the default when running on the Windows XP platform. This look and feel automatically appears if your app is using Swing's WindowsLookAndFeel class (either by way of UIManager.getSystemLookAndFeelClassName() or by explicitly using com.sun.java.swing.plaf.windows.WindowsLookAndFeel) on a machine running the Windows XP operating system. The following example illustrates the preferred way of setting the look and feel to match that of the native platform:

```
UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
```

If you prefer your Swing app to look like a classic Windows app you can turn of the XP look using the system property swing.noxp=true. This is typically used as a command line parameter, e.g.:

```
java -Dswing.noxp=true -jar SwingSet2.jar
```

Note that the swing.noxp property may not be supported in future releases.
So, this option can be declared deprecated :-)

Release 1.4.2 introduces a look and feel based on GTK+ 2.0. In order to get this look and feel you need to explicitly ask for it by way of UIManager.setLookAndFeel("com.sun.java.swing.plaf.gtk.GTKLookAndFeel") or by defining the system property swing.defaultlaf to be com.sun.java.swing.plaf.gtk.GTKLookAndFeel.
For those who is going to use GTK+ 2.0 based look and feel I say that this look and feel is quite "raw" and has a number incompatibilities caused by GTK+ specifics and needs to be resolved. Read more about details here:
http://java.sun.com/j2se/1.4.2/docs/guide/swing/1.4/Post1.4.html

- **Question:** Today many mice have more than three buttons. And people program the fourth - thumb button differently. I want to catch an event that usually done by right click - popup menu. My friend uses for that a middle button on three buttons mouse. If somebody uses fourth (even fifth!) button for that how my program can know it?

Answer: If your program supposed to support all kind of mice use boolean isPopupTrigger() method of MouseEvent class. This method returns whether or not this mouse event is the popup menu trigger event for the platform.

Note: Popup menus are triggered differently on different systems. Therefore, isPopupTrigger should be checked in both mousePressed and mouseReleased for proper cross-platform functionality.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** How can I minimize "Flicker" in animation? Solution 1:

  Answer: Solution 1:

  Override update() : Flickering in animation occurs because default update() method clears the screen of any existing contents and then calls paint(). To reduce flickering, therefore, override update(). Here is how just add the following code to your applet:

```
public void update(Graphics g) {
   paint(g);
}
```

  What the update is now doing is just calling paint() and not clearing it, a further refinement to the above will be to override update() method and painting only the region where the changes are taking place. Here is how:

```
public void update(Graphics g) {
   g.clipRect(x, y, w, h);
   paint(g);
}
```

  Solution 2 will be described tomorrow

- **Question:** On some Windows 2000 machines our application (uses Java 2D) causes system crash - blue screen of Windows death :-)...

  Answer: It can happen if you J2SE 1.4.1 and later. The problem is not in Java itself but rather in obsolete video drivers that can contain bugs. Java 2D functionality in J2SE 1.4.1 and beyond is implemented using DirectDraw and Direct3D for various off screen rendering functions.
  Possible solution is to update video driver. Find out what is chipset your graphical card uses and then go to the site of manufacturer, for example
  http://www.atitech.com

  If it does not help you can run your application with different flags.

  One example of a known problem is that using Direct3D on some small subset of Windows 2000 systems has been known to cause a system crash. You can also try launching your application with the following command-line flag set:

```
-Dsun.java2d.d3d=false
```

  This forcibly disables our use of Direct3D and avoids any Direct3D-specific problems. You can also try the following flag:

```
-Dsun.java2d.ddoffscreen=false
```

  This flag disables our use of DirectDraw and Direct3D for off screen images (such as the Swing back buffer).
  If a problem persists, try launching the application with this flag:

```
-Dsun.java2d.noddraw=true
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

This flag disables our use of DirectDraw and Direct3D completely and thus avoids any problems you may be seeing associated with our use of those APIs and their respective drivers on your system.

This tip is based on release notes for J2SE 1.4.2:

http://java.sun.com/j2se/1.4.2/relnotes.html#vm

- **Question:** How can I minimize "Flicker" in animation? Solution 2.

  Solution 1 was described in our tip yesterday
  Solution 2:

  Use double-buffering: double buffering is the process of doing all your drawing to an off-screen and then displaying the entire screen at once. It is called double buffering because there are two drawing buffers and you switch between them. Use double buffering only if the above solution alone does not work. The following code snippet describes how to do it.

```
Image offscreenImage;
Graphics offscreenGraphics;
offscreenImage = createImage(size().width, size().height);
offscreenGraphics = offscreenImage.getGraphics();
offscreenGraphics.drawImage(img, 10, 10, this);
g.drawImage(offscreenImage, 0, 0, this);
```

- **Question:** I am trying to construct my GUI so that my images can act like JButtons. Can I do this?

  Answer: You could try JButton.setIcon()... An iconimage can be created with something like:

```
Icon myicon = new ImageIcon(mybufferedimage);
JButton mybutton = new JButton(myicon);
```

  then add your button to the appropriate container...

- **Question:** I am loading an image into my application from users and do not know in advance what dimensions are. How do I find a width and height if it is an image and how to know if it is not? Maybe it is text file or some another kind of no image file.

  Answer: The AWT provides two ways for you to track image loading: the MediaTracker class and the ImageObserver interface.

  The MediaTracker class is a utility class to track the status of a number of media objects. Media objects could include audio clips as well as images, though currently only images are supported. The MediaTracker class is sufficient for many programs. You just create a MediaTracker instance, tell it to track one or more images, and then ask the MediaTracker the status of those images, as needed.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The ImageObserver interface lets you keep even closer track of image loading. Two methods: getWidth and the getWidth method of the Image class take one argument ImageObserver and return dimensions or -1 if it is not an image.

- **Question:** What is a rasterizer? Why doesn't Java 2D use a native rasterizer for rendering fonts?

  Answer: A rasterizer takes ideal shapes and produces coverage values for each pixel. The coverage values represent how much of each pixel is covered by the shape. These coverage values are called alpha values. Each pixel has its own alpha value which indicates the transparency of the pixel.
  The Java 2D API uses the T2K font rasterizer from Type Solutions. The Font Fusion font engine is independent of processor and OS and compatible with all industry standard font formats. It is a full featured lean, mean and fast next generation small-footprint multilingual scaleable outline font technology. Font Fusion was designed for both gray-scale and black and white, ROM and non-ROM, based devices where the fonts may reside locally or remotely.
  By allowing T2K to rasterize fonts, the Java 2D API isn't dependent on native rasterizers for the scalable outline fonts that it supports, which results in more consistent metrics display across different platforms and between on-screen and off-screen rendering.
  --
  this tip is based on SUN's FAQ's

- **Question:** On Solaris, Java Advanced Imaging complains about lack of access to an X server...???

  Answer: Java Advanced Imaging versions previous to JAI 1.1.1 used the AWT toolkit to load GIF and JPEG files. This problem is a manifestation of a JDK bug in which creation of the AWT Toolkit class results in an attempt to open the X display. To work around this problem in Java Advanced Imaging versions prior to 1.1.1, either make an X display available to the Java runtime using the DISPLAY environment variable (no windows will appear on the display), or consider running a dummy X server that will satisfy the AWT, such as the Xvfb utility included with the X11R6.4 distribution.

  In the JAI 1.1.1 version, the GIF and JPEG decoders were improved to no longer have a dependency on the X server.
  --
  this tip is based on SUN's FAQ's

- **Question:** How can I call swings in browser?

  Answer: You can download "HTML Converter Version 1.3.0_01" from SUN and your browser will use Java plugin instead of default browser's JVM:
  http://java.sun.com/products/plugin/1.3/docs/htmlconv_01.html

  or you can download swing.jar separately and add
  "archive=swing.jar" to your applet tag.

  Tips from http://JavaFAQ.nu

- **Question:** I need "Always On Top" functionality for my GUI...

  How could I do that?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: One of the main points of Java is to be completely platform independent. However, sometimes it will occur that the developer of an application will know that his or her application is only going to be run on a specific platform, for example, Win32.

Below are the steps for writing a Java application that makes a Win32 API call. The application generates a Swing JFrame and makes it system modal, or gives it the "Always On Top" functionality, similar to that of the Windows NT Task Manager:

1. Write the Java code for the application
2. Run javah.exe on your .class file to generate a C header file
3. Write the implementation of your native methods
4. Create the shared library
5. Run the application

The full text of this article is here:

http://community.borland.com/article/0,1410,20679,00.html

- **Question:** I want to disable the users ability to close a window (JDialog) using the "x" button at the top right corner of the window. Can I do that?

  Answer: It is simple :-)

  ```
  myDialog.setDefaultCloseOperation(WindowConstants.DO_NOTHING_ON_CLOSE);
  ```

- **Question:** I want to erase text in a TextArea. I want to use some function clear(), clearAll(), erase() but there are not in API... How someone supposed to do it?

  Answer: It is very simple, just use textArea.setText("");

- **Question:** A couple of canvases inside my applet using BorderLayout... How I can find out their sizes? getSize() seems to only work with the Applet class and not the Canvas class.

  Answer: Applet is a Container, and do like this:

  ```
  Component[] components = applet.getComponents();
  for (int i = 0; i < components.length; i++) {
    Dimension size = components[i].getSize();
  }
  ```

- **Question:** I am trying by one statement to change the font for all Swing components that are on JFrame

  ```
  myFrame.getContentPane().setFont( sysfont );
  ```

  But it does not help...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Until you define your own look and feel style there is no way to do it with command... Try to do it by using UIManager, something like this (for labels, for example):

```
UIManager.put("Label.font",new Font("Arial",Font.ITALIC,10));
```

The line above will change font on all labels. You can do it with other components. Just remember that some components can have another keyname, not like "component.font"...

- **Question:** I use printing of Swing components. It is slow... Can I increase the speed?

Answer: Yes, you can. Turn double buffering off. Although this feature increasing graphical performance and set on by default, it adds unnecessary job for your printer...

```
RepaintManager.currentManager(this).setDoubleBufferingEnabled(false);
```

Probably you can restore it after printing is over.

- **Question:** Is there a way to detect if the CAPS LOCK key is active?
I want to detect it even if it was activated before the application started.
I can detect if the user press the key, but not if the key has been pressed before.

Answer: Just use

```
java.awt.Toolkit.getDefaultToolkit().getLockingKeyState(java.awt.event.KeyEvent.VK_CAPS_LOCK);
```

- **Question:** Do you know how to make windows button on taskbar flash?

Answer: It works only for Windows ME/2000/XP.
Use Dialog window! Whenever you create a new Dialog OS (Windows) the task bar will flash the corresponding button. It is a property of current OS, not Java... That's why it will not work on Unix systems that have now task bars as well, for example with Gnome GUI.

- **Question:** I have read that Java Advanced Imaging API uses native code... Does it mean that this part of Java is not 100% pure Java?

Answer: The Java Advanced Imaging API may be run without any native code, i.e., without the code which provides native acceleration. When run in this manner, the Java Advanced Imaging API uses only Java code.
Native code is supplied for performance purposes only. A C library is supplied for SPARC, Linux and Intel x86 platforms which accelerates most of the operators. In addition there is a lower level SPARC library (VIS) and a MMX library allowing additional hardware acceleration for many operators. If Java Advanced Imaging code does not find these libraries, pure Java code is used.
However, the Java Advanced Imaging API has not yet been certified as meeting the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

requirements of the "100% Pure Java" Program per se.

- **Question:** JLabels: when I add "too much" text I can't see all of it.
Also, how to break line into a few pieces? Possible solutions?

Answer: The getPreferredSize() method of the JLabel will return you the minimum size that required to display the text in full. But you need to choose a layout that will not respect those sizes. Not so easy question as well :-)

It was suggested to use JList instead of JLabel and disable it with setEnabled(false). The selection will be disabled and you need to add your lines as lines to data model of JList. Probably you need to take care about UI properties of JList (copy fro JLabel) to make it looking the same...

- **Question:** Is there a Java class or component or something like to browse the web inside java Application?

Answer: I know two and they are not free... If you know free component please send me info and I will publish it here.

HotJava browser from SUN is no longer supported and end date is April 2003.

1. Clue Web Browser Components:

Clue Web Browser is the only browser solution designed to realize this new web browser concept. With its modular and scalable architecture, Clue Web Browser can be integrated with any application and run on a wide range of devices, such as PDAs and set-top boxes. Through our relentless pursuit of perfection, Clue Web Browser exhibits a solid implementation of the latest web technologies with excellent performance. It offers rich functionality and flexibility for a wide variety of web-related applications. To meet the specific requirements of different devices, we offer various editions for desktop and embedded devices.

2. WebWindow 2.0 is a scalable Java browser component. With the WebWindow you can take advantage of Java's "write once, run anywhere" design and run your application on multiple operating systems and device platforms with a wide range of resource limitations. With a small core footprint, WebWindow can be used on devices ranging from PDAs, settop boxes, enterprise servers, and native web browsers all with little or no code changes to the core browser technology.

By using the WebWindow in your application, you can avoid compatibility issues between different browsers and you can have a truly "100% Java" product. WebWindow can be utilized in a number of ways in an application ranging from a simple help file displayer to a full fledge UI framework for building dynamic GUIs.

Question: How can my application dispose or close a Frame from another Frame AWT?

For example I press button and another frame disappears...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: easy!
Event handling procedure add following code

for example,

```
if(e.getSource() == buttonOK){
   frameName.dispose();
}
```

or

```
framename.setVisible(false);
```

Remember, last method will still keep frame object alive and waste memory.

And dispose() method releases all of the native screen resources used by this Window, its subcomponents, and all of its owned children. That is, the resources for these Components will be destroyed, any memory they consume will be returned to the OS, and they will be marked as undisplayable.

- **Question:** The following app works fine, but when I start it, I cannot close it using the X at the right top of the form...
Please help me on the following. I'm just starting to use java (JDK1.3). The following app works fine, but when I start it, I cannot close it using the X at the right top of the form. What should I add to fix this problem? The source is shown below.

```
import java.awt.*;
import java.awt.event.*;

public class MyApplication extends Frame{
  public static void main(String args[]){
     Frame f = new Frame("MyApp");
     f.setLayout(new FlowLayout());
     f.add(new Button("A"));
     f.setVisible(true);
  }
}
```

Answer: You should add a listener to handle the closing of the window when the X box gets clicked on.

```
f.addWindowListener(new WindowAdapter(){
  public void windowClosing(WindowEvent evt){
     System.exit(0);
  }
});
```

Read on this, and other ways to do it, in the sections dealing with event handling, in whichever Java text you are using.

- **Question:** How can I set a JFrame to be full screen at the start of a program? I want no borders, no titles and I just want to use the entire screen.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Try using JWindow instead, that one can be customized to have no borders or titles... as for size, I think you can use

```
setBounds(GraphicsEnvironment.getLocalGraphicsEnvironment().
getDefaultScreenDevice().getDefaultConfiguration().getBounds());
```

to fill out the entire screen.

- **Question:** Why can not I mix AWT and Swing?

Recently, I have been hearing a lot of people from various newsgroups and website saying, java swing and AWT can't be in the same application. They will not work together and they might produce unexpected results. At the same time, I don't hear people saying "why" you shouldn't use swing and AWT together.

Could someone out there shed some light for me? Is their any logical reason why we shouldn't mix swing and AWT in the same application/applet? If there is a problem mixing swing and AWT ... what are the results, what can happen? I design using IBM's Visual Age for Java 3.0, and I mix swing and AWT in the same application/applet, it works fine when testing in the IDE (I haven't tested it outside of the IDE yet). If you have tested application/applets outside of the IDE, please let me know what happened?

Answer: There are fundamental incompatibilities in the way they draw themselves. AWT java classes are not "pure" Java classes, they use underlying C/C++ native code (dependable on operation system) that can cause different appearance in different OSs.

Swing is pure Java implementation and has no native code at all. Swing applications look the same.
> If there is a problem mixing swing and AWT... what are the results, what can happen?

Some objects drawn on top of others are not properly occluded. This is most obvious with drop down menus, which have a tendency to stay visible even after you have selected a menu item. Another problem is that if you use AWT components on a JTabbedPane they will not disappear when you switch tabs. There are many similar issues.

- **Question:** Again about difference between AWT and Swing

I have a **Question:** What are the architectural differences between Swing and AWT??

Answer:
There are no big architectural differences, the class hierarchy is almost the same. The reason is that Swing is built upon AWT.
The most significant difference is how the components are drawn to the screen. AWT is so called heavyweight components and have their own view port which sends the output to the screen. Swing is lightweight components and does not write itself to the screen, but redirect it to the component it builds on. Heavyweight components also have their own z-ordering. This is the reason why you can't combine AWT and Swing in the same container. If you do, AWT will always be drawn on top of the Swing components.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

You can combine AWT and Swing, just don't do it in the same container (e.g. panel, group box, etc.) and don't put a heavyweight component inside a lightweight.

Another difference is that Swing is pure Java, and therefore platform independent. Swing looks identically on all platforms, while AWT looks different on different platforms.
--
by Odd Vinje

- **Question:** I have a JFrame for customer registration with a lot of input fields. In this screen you can create a new customer, get customer with specified customer number and you can update a customer.

  In this JFrame is it possible to clear all fields without specifying each field?

  Answer: This snippet is for TextFields, you should be able to adapt for JtextFields very easily.

```
public static void resetTextFields(Container c) {
  Component [] components = c.getComponents();
  for (int i = 0; i < components.length; i++ ) {
     if (components[i] instanceof Container)
        resetTextFields((Container) components[i]) ;
     else if (components[i] instanceof TextField)
        ((TextField) components[i]).setText("") ;
  }
}
```

  Real Gagnon from Quebec, Canada
  * Looking for Java or PB snippets ? Visit Real's How-to
  * http://www.rgagnon.com/howto.html

- **Question:** Swing is "lightweight" components. Its graphics primitives are implemented in 100% Pure Java. How does it draw pixels on screen?
  Does it still rely on any native code to access the frame buffer?

  Answer: We mentioned before in our tips that Swing components is 100% pure Java. It isn't fully correctly if we are speaking about containers on which they can draw. Swing is still based on AWT, and even Swing components must have at least one heavyweight container. In other words, JFrame, JApplet are *not* lightweight.

- **Question:** I need to put in a JtextField characters like Cyrillic ...
  How can I do this?

  Answer: You use the Unicode string \u???? corresponding to your character.
  A table of codes can be found at www.unicode.org/charts/
  --
  by Charles Megnin

- **Question:** Is there any way to accept password from a java application. As it is a password, IT SHOULD NOT BE SHOWN(ECHOED) WHILE IT IS ENTERED.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: If you are using swing, you can use JPasswordField, and set your echo character.

- **Question:** How can I change the default icon on application window (java cup) to my own?

  Answer:
  window.setIconImage(Toolkit.getDefaultToolkit().getImage("image.gif"));

- **Question:** How can I read a Tab delimited text file into a JList?

  All I need is the first item in each record. I import each line with a BufferedReader.Readline () and then look at each char with record.substring(n,n+1). How can I check if the next char is a Tab?

  Answer: StringTokenizer st = new StringTokenizer(record, "\t");
  while (st.hasMoreTokens()) {
  String token = st.nextToken();
  // ... do something with the token
  }

- **Question:** How do I generate mouse click event but without a real click?
  Could you tell me how I can generate an event (like a mouse click) from a Java application? (I want generate a mouse click event but without a real click on the mouse).

  Answer: See "simulate keyboard pressing" below: You can use java.awt.Robot since java1.3

- **Question:** Why would anyone use AWT and not Swing today?

  Answer: There are several handheld devices that do not have a lot of memory or CPU power available, AWT is more suitable for them (Smaller footprint).

- **Question:** Need I to remove all the listeners on all of the components in the Frame when I close window of the Frame?
  I've got a Frame, which has in it a bunch of components, some of which themselves may be containers. Many of the components in this frame have listeners.

  When somebody closes the window of the Frame, I want to completely get rid of the Frame, so that the garbage collector will later clean it up, freeing its memory. However, I haven't yet figured out a way to do this without tracking every single component by myself. Just calling dispose() on the main Frame doesn't seem to be good enough. Calling «RemoveAll()", even recursively, doesn't seem to be good enough.

  I *suspect* that the problem may be that I have to remove all the listeners on all of the components in the Frame. Is there a way to get a list of the listeners out of a given component? I really don't want to have to track every single button I add to every single Frame, and every single Listener I add to every single component. If I have to keep track of all of that, it sort of defeats a lot of the advantages of a well-defined object oriented system.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: I think you're slightly confused here. When another object registers as a listener for a component within the frame, it's the component within the frame that is holding a reference to the external object in its listeners list - not vice versa.

This means you don't really need to explicitly remove every listener that any other object registered on components in the frame.

What you need to worry about is however the listeners that the frame itself registered with some other components that is still alive. If the frame gets disposed without removing these, the objects that were being listened to will retain a reference to the frame and this can cause the frame to stay around as long as these objects which hold these references stay alive.

So look for all Listeners that your frame registered itself as on components external to itself (should be fairly easy to see since you normally do it in your constructor) and remove those before disposing off the frame.
--
Kala

- **Question:** Main disadvantage of Java GUI is that it is often slow. But I have seen also very fast working GUIs. Unfortunately the code is hidden and I do not know the technique for writing of fast Java GUIs.

Answer: I can describe one of main technique you can use. It does not give you full solution, but will certainly speed up your GUI.

The main idea is to use "lazy" initialization. Instead of creating and initializing of all GUI components in constructors during start up time postpone it to later time until you really need. Let say, you have a lot of tab panels with many elements on each tab panel.

Your constructors should be "quite" empty and do not create and initialize those small elements until your tab panel is chosen. You should have very small constructor for tab panel itself and additional lazy constructor for the rest. It should be called when user clicks on that particular tab.

This does not decrease the full time of initialization, but spreads it up. The user will not feel big delays during start up of your application. You should use these lazy constructors just before you are going to call any paint method.
--
AP. (J.S.)

- **Question:** I've a question where skillful java programmers sure will laugh about, but I ask nevertheless.

What exactly means >>> Graphics g <<<? How must I use it and why is it so important to use?

Answer: It basically says: "where should I paint? On the screen, on some RAM, on a printer buffer?"

It is sort of a conduit to that medium.
Graphics class is the set of tools, which allows you to set background, colors, fonts;

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

draw lines, circles, arcs, polygons etc to a component. Here is the definition from the class javadoc documentation:

"The Graphics class is the abstract base class for all graphics contexts that allow an application to draw onto components that are realized on various devices, as well as onto off-screen images.
A Graphics object encapsulates state information needed for the basic rendering operations that Java supports.
This state information includes the following properties:
The Component object on which to draw.
A translation origin for rendering and clipping coordinates.
The current clip.
The current color.
The current font.
The current logical pixel operation function (XOR or Paint).
The current XOR alternation color (see setXORMode(java.awt.Color)). "
--
R. Green, Gilgames

- **Question:** I'm resetting the TitleBorder's title via setTitle, but no change occurs....

I'm trying to make a dialog box containing a panel whose TitleBorder's text dynamically reflects the function of the list box within it. The title text and the list box contents should change according to the buttons pressed elsewhere in the dialog box. I'm resetting the TitleBorder's title via setTitle, but no change occurs. Do I need to update the dialog? If so, how do I do this?

Answer: Yes, you need to update the dialog.
How? Call the validate() and invalidate () methods

- **Question:** Every time you create a dialog-object, the dialog pops up in the upper left corner of the screen. Now I want it to be automatically placed in the center of the application's frame.

Answer: private void center() {
Dimension screenDim = Toolkit.getDefaultToolkit().getScreenSize();
setLocation(((screenDim.width - getSize().width) / 2),
((screenDim.height - getSize().height) / 2));
}

You can call this anytime after calling pack() on the JDialog, before setVisible().
--
Dave Neuendorf

- **Question:** How can I disable the right click on the label?
I have a JTree and a JPopup: I have forced the code to be able to select the node with a right click and show the popup. The problem is that i can right click on the popup labels to select them, but I'd like to select them only with a left click. How can I disable the right click on the label? Can somebody help me please?

Answer: You can add in an if loop that only executes if the left mouse button is down using code like the following:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
MouseEvent e;
if ((e.getModifiers() & e.BUTTON1_MASK) != 0)
{ // code for left button click }
```

and just so you know,
InputEvent.BUTTON2_MASK is the middle or scroller button
InputEvent.BUTTON3_MASK is the right button in windows
--
by MSW

- **Question:** How do I count the width in pixels of the text before drawing them into graphics?
I have the Font and the String and I am using them in an applet:

```
graphics.setFont(font);
graphics.drawString(str);
```

How do I count the width in pixels of the text before drawing them into graphics?

Answer:

```
import java.awt.*;

Toolkit tk = java.awt.Toolkit.getDefaultToolkit();
FontMetrics fm = tk.getFontMetrics(new Font("Arial", Font.PLAIN, 10));
int width = fm.stringWidth("Your String");
```

--
Jarkko

- **Question:** If I do (CustomFrame extends Frame) something like this:

```
Frame newFrame = new CustomFrame();
```

what are the advantages of doing this over:

```
CustomFrame newFrame = new CustomFrame();
```

Answer:
There is one difference.
If the reference is a Frame type you can always downcast it, but not if it is a CustomFrame type.
Might be nice sometimes.
--
Soren ' Disky ' Reinke

- **Question:** How can I have a filtered copy and paste in JTextPane? I am trying to copy and paste styled text and remove all the new line '\n' characters from the copied text. So when I paste the text in must still have its styles as well as no '\n' characters.

Answer: You could override the paste() method of JTextPane. (This is inherited from JTextComponent). You can look at the source code of JTextComponent to see what it does currently and what modifications will be needed for the modified action.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

--
BK

- **Question:** I want to know what is the difference between JScrollPane and JScrollBar...is both same or not..?

  Answer: No. A JScrollPane is a container for components which can grow larger than the visible area that can be displayed (i.e., a JTable or a JTree, for example). When you place a component like these into a JScrollPane then the visible area is displayed and appropriate scrollbars presented to allow scrolling to non-visible areas.

  A JScrollBar is just that, a scroll bar.
  Read the Java docs for both classes and the differences should be most apparent.
  --
  Darryl L. http://welcome.to/mcpierce

- **Question:** Is it possible to change delays that affect appearing, keeping and disappearing of tooltip?

  Answer: It was difficult to find the answer but finally I found in "Swing" book that is free to you on our site.
  The ToolTipManager is a service class that maintains a shared instance registered with AppContext. We can access the ToolTipManager directly by calling its static sharedInstance() method:

  ToolTipManager toolTipManager = ToolTipManager.sharedInstance();

  Internally this class uses three non-repeating Timers with delay times defaulting to 750, 500, and 4000. ToolTipManager uses these Timer's in coordination with mouse listeners to determine if and when to display a JToolTip with a component's specified tooltip text. When the mouse enters a components bounds ToolTipManager will detect this and wait 750ms until displaying a JToolTip for that component. This is referred to as the initial delay time.

  A JToolTip will stay visible for 4000ms or until we move the mouse outside of that component's bounds, whichever comes first. This is referred to as the dismiss delay time. The 500ms Timer represents the reshow delay time which specifies how soon the JToolTip we have just seen will appear again when this component is re-entered.

  Each of these delay times can be set using ToolTipManager's setDismissDelay(), setInitialDelay(), and setReshowDelay() methods.

  ToolTipManager is a very nice service to have implemented for us, but it does have significant limitations. When we construct our polygonal buttons we will find that it is not robust enough to support non-rectangular components.

- **Question:** How can I test if a JTextField is empty? My code is below but does not work.

```
if (myJTextField.getText() == " ")
    return true;
else
    return false;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: You have to compare String content, not String handler...so you have to use the equals() method.

For example:

```
if (myJTextField.getText().equals(""))
    return true;
else
    return false;
```

Or

you can test whether text = = null,
then check if myJTextField.getText().length() == 0
if it is not, do text=text.trim() to remove spaces,
then check text.equals("") to see if it is an empty String
--
Lucio Benfante, Arnaud Berger and Jorge Jordao

- **Question:** What's the difference between a Frame and a canvas? Can someone tell me the difference?

Answer: A Frame is , well , a frame - a window, with borders, close/minimize/maximize buttons (usually), and can not contain other elements, such as a menubar ,buttons panels, etc.

A Canvas is a user interface element (which you can place in e.g. a frame) onto which you can draw/display something..
--
by Nils O. Selasdal

- **Question:** I know how to make a text box using Visual Basic that can only accept certain keys for example if I wanted to make a box where you can type only numbers in it or letters. However, when it comes to Java I have not been able to figure this out. Is there a way to do this and make a textbox where you can only type letters in it?

Answer: First of all, 'ASCII' is not a synonym for 'text'. ASCII is just one of many ways of encoding text to numeric values. It is 7-bit and only includes values from 0 through 127.

Now, on to your question. With ASCII, 'numbers' and 'letters' are a little easier to check, since the only letters are 'A'-'Z' and 'a'-'z'. However, Java uses Unicode and has all sorts of fun extra characters.
But you do have a few handy methods on the java.lang.Character class such as isDigit and isLetter

Now, if you are using swing, then to achieve what I'm guessing you want is quite simple. Just hook a custom text document to the JTextField you're using and have it do the filtering.

** Here's the part of The Java Tutorial that goes over it
http://java.sun.com/docs/books/tutorial/uiswing/components/textfield.html#validati

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

on

Here's someone's Swing FAQ on that
http://users.vnet.net/wwake/swing/faq.html#text_check

Here's a JavaWorld article that describes how to do that to limit the length of input. Shouldn't be too hard for you to change to your specific needs.

http://www.javaworld.com/javaworld/javaqa/2000-05/02-qa-0512-textbox_p.html
--
John A.Cruz

- **Question:** How can I force Java not to draw the background?
I have the following problem: I display a .gif image on a panel with the g.drawIcon() method. If I now display the image of a car on a track then the background of the image (black) gets displayed, too.

  How can I force java not to draw the background?
I tried the setXORMode(Color col) method which filters the given color out but this ruins my image.

  Answer: Just make your GIF image have a transparent background. Java already handles drawing GIFs with transparency. The black color in your car image is not the transparency color of your image.
--
dalal

- **Question:** I was wondering how to create a Directory Dialog, just like the FileDialog class, but for selecting directories instead of files.

  I'm sure this has been done, and I apologize if it is in a FAQ. Just point me in a direction, and I will go. Swing or AWT is fine.

  Answer: In Swing, you can use JFileChooser and set it up to choose a directory. See the API docs for details.
--
Chris Smith

- **Question:** Can any one please tell me whether I can create a Window/JFrame in Java that is not rectangular in shape?

  If the answer is yes, then where can I get the information to do that?

  Answer: Unfortunately answer is "No"...

- **Question:** I'm trying to do some image manipulation (mainly resizing images) in a Linux Servlet/JSP environment, to do that I use the Java 2D API.

  This works fine on my Windows development PC, but does not work on an X-less Linux hosting platform. My hosting company is not running X and has no plans to do so.
So, my question is, it is possible to use the Java 2D API on a Linux box without an X server running?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer Part 1: A frequently asked question in the context of loading images is that of the missing X server. UNIX users who just want to load an image via the AWT without displaying it get an error message if there is no X server installed. In fact, the X server is required for that.

java.awt.Graphics methods such as drawLine (), fillOval (), drawString (),... are implemented in the default JVM with native graphical functions (except in some cases for Java2D) : That means that drawLine () finally calls a GDI system function on Windows or X11 function on a X11/UNIX machine even if the drawing is done in an off-screen image using the class java.awt.Image. This ensures the best performance for drawing graphics with Java.

When no X11 Display is available on UNIX machines or when GDI resources are low on Windows, it is impossible to compute off-screen images with java.awt.Graphics methods, even if your program doesn't need to display these images. Typically, this situation happens for Servlets returning dynamically generated images like pies, charts or web counters.

It is also impossible to compute off-screen images when the Java security manager forbids access to any Toolkit or the AWT library.
Possible solutions will be published tomorrow! (end of Part 1)

- Part 2 (Part 1 in yesterday's tip):
Possible solutions:

1. Install a virtual frame buffer X Windows System server for XFree86. XVFB (X Virtual Frame Buffer) is an X server that is able to run on machines with no display hardware and no physical input devices.

XVFB emulates a dumb frame buffer using virtual memory. XVFB doesn't open any devices, but behaves otherwise as an X display. XVFB is normally used for testing servers. Using XVFB, the mfb or cfb code for any depth can be exercised without using real hardware that supports the desired depths. XVFB has also been used to test X clients against unusual depths and screen configurations, to do batch processing with XVFB as a background rendering engine, to do load testing, to help with porting an X server to a new platform, and to provide an unobtrusive way of running applications which really don't need an X server but insist on having one.

2. Use PJA Toolkit available for free at:

http://www.eteks.com/pja/en/

3. Use Java 1.4 which is available now. This version can work without X-server.

- **Question:** How do I print a frame? I need this because I'm doing some graphics with java, and need to print them on paper. I'm using the AWT.

Answer: You would rather paint your graphics on a panel/canvas/JPanel than on a Frame/JFrame, I suppose, and the best would be to use the 2D-Print-API for this, there are a lot of example at Sun available and I have some at my own WebSite, if you need some examples:

http://java.sun.com/docs/books/tutorial/2d/printing/

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

http://www.jalice.net/discussion.htm

If you are restricted to use the java.awt.PrintJob, you can use something like this:

```
PrintJob myJob = getToolkit().getPrintJob(MyClass.this, "My Print", null);
if (myJob != null) {
   Graphics printgraph =
  myJob.getGraphics();
   if (printgraph != null) {
      printAll (myJob);
      printgraph.dispose();
   }
  myJob.end();
}
```

--
Linda Radecke

- **Question:** I'm currently developing my first actual graphical application in Java, using Swing. I feel that I've got a pretty good grip at object-oriented design and analysis, but I'm somewhat uncertain when it comes to connecting the user interface with the application logic.

  Should the GUI components be in the main application class, separated into several classes depending on their position on the screen, or included in the "application logic"-classes? What about the different EventListener I need?

  Answer: The biggest, biggest mistake people frequently make is to embed application logic right in with their GUI code. It is almost inevitable with most programmers, unfortunately.

  One idea is to use top-level packages, perhaps GUI, kernel, and database.
  In the GUI package, have one package for each frame or main thing, like gui/main, gui/login, gui/help, gui/detail, etc. From all the many tiny pieces of gui code that see button pushes, get events, enable or disable things based on user authorization level, etc, do *NOT* embed application logic there.

  Make all these tiny pieces of gui code call a separate NON-GUI arbiter interface (not class) that has methods like:

  * void initialize ()
  * void beginSearch (String phoneNumber)
  * void login (String user, String password)
  * void help (String subject)
  * void exitApplication ()

  The GUI calls an interface (the arbiter), and the arbiter calls a simple interface to the GUI. This is one scheme that separates the GUI from application logic.
  --
  Max Polk

- **Question:** I would like to display images of various sizes using JInternalFrame.
  1. I want to create a JPanel and paint the image on this JPanel.
  2. Create a JScrollPane and add the JPanel to the JScrollPane.
  3. Add the JScrollPane into the JInternalFrame.
  4. I want to set the size of JInternalFrame to fit the image size.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

I am having trouble with the setSize () method of JInternalFrame. If I don't use this method, the JInternalFrame doesn't appear. But I cannot seem to find a good way to set its size to match the image size.

Answer: Basically you have three ways to set the size of JInternalFrame. You can use either:

setSize(),
pack(), or
setBounds().

When you call pack () instead of setSize, the JInternalFrame calculate the size based on its content inside.

One way would be to use ImageIcons wrapped into JLabels, add them to your component and call pack() on the JInternalFrame instead of a fixed size.

As an alternative way instead of overriding the paintComponent() of your JPanel, what you could do as well, of course and decide whether you like to draw the image on it's original size, scaled or tiled. But you will then need to determine in turn the size of your JPanel, IMO. Thus, I think, it might be the better approach to use ImageIcons instead.
--
Linda

Question: While compiling I get an error that says something about a JPanel that is accessed from within inner class and needs to be declared final. That's OK. It can be as final as it wants to. BUT...

What does it mean when a JPanel is final?
and
Why in the name of the Lord does the compiler want it to be declared as final?

Answer: A JPanel isn't ever final. The compiler is telling you that the variable referencing the JPanel must be final. When a variable is declared as final, it means that the variable cannot be pointed at some other object (or contain some other value in the case of primitives).

For instance, this code would produce an error:

```
final int foo = 42;
foo = 37;
```

When using inner classes, any local variables which are used inside the inner class must be declared final. The compiler requires that the reference be final because the inner class makes a copy of the local variable value so that it can use it. This is fine as long as the two copies of the value don't get 'out-of-synch'.

By requiring that the variables be final, the copies are guaranteed to always point to the same object (or contain the same value for primitives).

See the Inner Class Specification for details:
http://java.sun.com/products/jdk/1.1/docs/guide/innerclasses/spec/innerclasses.doc

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

.html
--
Jim Sculley

- **Question:** I need some picture icons for my application that I'm building. Could someone point me to some place where I could get some pictures?

  Answer: We have the copy of Java TM Look and Feel Graphics Repository on our site here:
  http://javafaq.nu/java/graphics/java-look-and-feel.shtml
  --
  John

- **Question:** I need to create image from two smaller images (two parts of map) and save it. I tried to do it with JAI, but I was able only to paste images over each other.

  Answer: The usual way of doing this is creating BufferedImage and painting those two images on its Graphics:

  1. You create empty BufferedImage of size of total size your two source images.
  2. Get graphics
  3. Draw your source images in appropriate location
  4. Enjoy result

  --
  Dmitry R

7. General Java

- **Question:** I was looking at java.util.Timer class and there is no possibility to cancel an individual task. I have to cancel whole Timer with all my scheduled tasks... What is solution?

  Answer: The solution is in your question :-) you could develop a subclass of Timer that stores a list of all scheduled TimerTask instances. Then retrieve the instance that you want to cancel:

  ```
  TimerTask [] tasks = timer.getScheduledTasks();
  tasks[0].cancel();
  ```

  tip based on SUN's forum answer

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Question: Why internationalization often named as i18n?

Answer: The term internationalization is abbreviated as i18n, because there are 18 letters between the first "i" and the last "n."

- **Question:** I often hear something like this "daemon runs" or "how to make my own daemon"...
  Is not it the same daemon and application? They are both executable. What is difference between daemons, services and applications?

  Answer: In UNIX terminology, applications that provide services rather than directly communicate with users are called daemons. For example network, printer services and so on...
  They do not need any direct interaction with user and run in background, invisible for GUI users.
  In Windows the same concept is known as a service.
  So, if you can not communicate with application that you develop and it is invisible - you can surely call it daemon or service, until all bugs are fixed :-)

- **Question:** How can I convert any Java Object into byte array?

  Answer: Very elegant way I found on SUN's web site:

```
public static byte[] getBytes(Object obj) throws java.io.IOException{
  ByteArrayOutputStream bos = new ByteArrayOutputStream();
  ObjectOutputStream oos = new ObjectOutputStream(bos);
  oos.writeObject(obj);
  oos.flush();
  oos.close();
  bos.close();
  byte [] data = bos.toByteArray();
  return data;
}
```

- **Question:** Why does the InputStreamReader class has a read() method that returns an int and not a char? Why we need additional steps for converting?

  Answer: When end of stream is reached we always get the value -1. It is not possible with char. How do we know when stream is empty? We can not use EOF, EOL and so on...

Question: What are direct and non/direct buffers (NIO)?

Answer: A byte buffer is either direct or non-direct. Given a direct byte buffer, the Java virtual machine will make a best effort to perform native I/O operations directly upon it. That is, it will attempt to avoid copying the buffer's content to (or from) an intermediate buffer before (or after) each invocation of one of the underlying operating system's native I/O operations.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

A direct byte buffer may be created by invoking the allocateDirect factory method of this class. The buffers returned by this method typically have somewhat higher allocation and deallocation costs than non-direct buffers. The contents of direct buffers may reside outside of the normal garbage-collected heap, and so their impact upon the memory footprint of an application might not be obvious. It is therefore recommended that direct buffers be allocated primarily for large, long-lived buffers that are subject to the underlying system's native I/O operations. In general it is best to allocate direct buffers only when they yield a measurable gain in program performance.

A direct byte buffer may also be created by mapping a region of a file directly into memory. An implementation of the Java platform may optionally support the creation of direct byte buffers from native code via JNI. If an instance of one of these kinds of buffers refers to an inaccessible region of memory then an attempt to access that region will not change the buffer's content and will cause an unspecified exception to be thrown either at the time of the access or at some later time.

Whether a byte buffer is direct or non-direct may be determined by invoking its isDirect method. This method is provided so that explicit buffer management can be done in performance-critical code.

- **Question:** I try to call the Runtime.exec() command by passing it an array of strings.

  The problem is that I need to use "&" in my command. It runs perfectly on shell prompt but not by Runtime.exec(). Java cuts all everything away after that sign, so my command is incomplete.

  Answer: No clear explanation to such behaviour. Use workaround:
  write your command into file (shell script) and then run this script. It will work :-)

- **Question:** I write a batch file dynamically.
  First, I collect all data into String value and then write " " as an end of line. But I can not execute this batch file because of this EOL.

  Answer: Use system defined line separator,

  String lineSeparator = System.getProperty("line.separator");

- **Question:** Why Java Hot Spot Compiler has the name - Hot Spot? Part 1.

  Answer: The best answer I actually found in "Technical White Paper for the Java HotSpot Virtual Machine, v1.4.1" at SUN web site:
  "Most attempts to accelerate Java programming language performance have focused on applying compilation techniques developed for traditional languages. Just-in-time (JIT) compilers are essentially fast traditional compilers that translate the Java technology bytecodes into native machine code on the fly. A JIT running on the end user's machine actually executes the bytecodes and compiles each method the first time it is executed.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Adaptive optimization solves the problems of JIT compilation by taking advantage of an interesting program property. Virtually all programs spend the vast majority of their time executing a minority of their code. Rather than compiling method by method, just in time, the Java HotSpot VM immediately runs the program using an interpreter, and analyzes the code as it runs to detect the critical hot spots in the program. Then it focuses the attention of a global native-code optimizer on the hot spots.

- **Question:** Why Java Hot Spot Compiler has the name - Hot Spot? Part 2.

By avoiding compilation of infrequently executed code (most of the program), the Java HotSpot compiler can devote more attention to the performance-critical parts of the program, without necessarily increasing the overall compilation time. This hot spot monitoring is continued dynamically as the program runs, so that it literally adapts its performance on the fly to the user's needs.

A subtle but important benefit of this approach is that by delaying compilation until after the code has already been executed for a while (measured in machine time, not user time), information can be gathered on the way the code is used, and then utilized to perform more intelligent optimization. As well, the memory footprint is decreased. In addition to collecting information on hot spots in the program, other types of information are gathered, such as data on caller-callee relationships for virtual method invocations. "

- **Question:** Why do I see the attribute in a derivative class? I can not access it anyway!

  Answer: You can access protected attributes in an extended class or within of package where this class is.

- **Question:** Are classes implementing Externalizable considered Serializable classes?

  Answer: Yes, since public interface Externalizable extends Serializable (according to API)

- **Question:** I tried to send Ctrl+Alt+Del key code to system from my class, but it does not work...

  Answer: The problem is that this combination is served differently from all "normal" key combinations on hardware level. When you press those three buttons an interrupt of processor occurs. When an interrupt occur processor looks at its number and executes special program in specific memory location. If you want to use this key combination you need to find proper windows library and use it through JNI. Java does not serve this special case...

- **Question:** The beta release of J2SE 1.5 is scheduled for late 2003. I have heard that there will be some changes in a code writing process that will make our life easier...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Yes, indeed. The improvements to J2SE being developed in project "Tiger" are intended to make programs clearer, shorter, safer, and easier to develop, without sacrificing compatibility. Java 1.5 will shift the responsibility for writing the boilerplate code from the programmer to the compiler...
Example:

Instead of writing
String title = ((String) words.get(i)).toUppercase();

you will do like this:

String title = words.get(i).toUppercase();
and it will decrease the number of compile errors. Compiler will try to cast to String without saying it explicitly.

More about all coming changes you can read in this excellent article:
"New Language Features for Ease of Development in the Java 2 Platform, Standard Edition 1.5: A Conversation with Joshua Bloch" on:
http://java.sun.com/features/2003/05/bloch_qa.html

- **Question:** I would like to create a class that can execute JScripts. Does any one know if there is such class?

Answer: Yes, it is Rhino. Rhino is an open-source implementation of JavaScript written entirely in Java. It is typically embedded into Java applications to provide scripting to end users.
Be aware that it has some limitations, Rhino is an implementation of the core language only and doesn't contain objects or methods for manipulating HTML documents.

More read about Rhino here:
http://www.mozilla.org/rhino/

- **Question:** When do I to choose an abstract class over an interface?

Answer: Choosing interfaces and abstract classes is not an either/or proposition...

"Often in a design, you want the base class to present only an interface for its derived classes. That is, you don't want anyone to actually create an object of the base class, only to up cast to it so that its interface can be used. This is accomplished by making that class abstract using the abstract keyword. If anyone tries to make an object of an abstract class, the compiler prevents them. This is a tool to enforce a particular design.

You can also use the abstract keyword to describe a method that hasn't been implemented yet-as a stub indicating "here is an interface function for all types inherited from this class, but at this point I don't have any implementation for it." An abstract method may be created only inside an abstract class. When the class is inherited, that method must be implemented, or the inheriting class becomes abstract as well. Creating an abstract method allows you to put a method in an

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

interface without being forced to provide a possibly meaningless body of code for that method.

The interface keyword takes the concept of an abstract class one step further by preventing any function definitions at all. The interface is a very handy and commonly used tool, as it provides the perfect separation of interface and implementation. In addition, you can combine many interfaces together, if you wish, whereas inheriting from multiple regular classes or abstract classes is not possible."

From "Thinking in Java", http://javafaq.nu/java/book/Chapter01.shtml#Heading29

- **Question:** What is difference between Hashmap and Hashtable?

  Answer: there is almost no difference, since "hashmap is Hash table based implementation of the Map interface" (citation from API). The difference is that Hashtable is synchronized.

- **Question:** My colleague, he is testing my code, got "Headless Exception". He jokes that is thrown by code which is written by headless people :-) He has shown me log file even. Is he joking with me?

  Answer: Probably not :-) I mean that you are Ok although this exception exists in reality.

  It is thrown when code that is dependent on hardware like keyboard, display, or mouse is called in an environment that does not support a keyboard, display, or mouse. For example, all modern servers have no display, keyboard and mouse.

- **Question:** I know that Hashmap implementation is not synchronized. How can I synchronize it?

  Answer: Really good answer often can be found in API!
  "If multiple threads access this map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally. (A structural modification is any operation that adds or deletes one or more mappings; merely changing the value associated with a key that an instance already contains is not a structural modification.) This is typically accomplished by synchronizing on some object that naturally encapsulates the map. If no such object exists, the map should be "wrapped" using the Collections.synchronizedMap method. This is best done at creation time, to prevent accidental unsynchronized access to the map:

  Map m = Collections.synchronizedMap(new HashMap(...));

  The iterators returned by all of this class's "collection view methods" are fail-fast: if the map is structurally modified at any time after the iterator is created, in any way except through the iterator's own remove or add methods, the iterator will throw a ConcurrentModificationException. Thus, in the face of concurrent modification, the iterator fails quickly and cleanly, rather than risking arbitrary, non-deterministic behavior at an undetermined time in the future.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Note that the fail-fast behavior of an iterator cannot be guaranteed as it is, generally speaking, impossible to make any hard guarantees in the presence of unsynchronized concurrent modification. Fail-fast iterators throw ConcurrentModificationException on a best-effort basis. Therefore, it would be wrong to write a program that depended on this exception for its correctness: the fail-fast behavior of iterators should be used only to detect bugs."

From Java API (v 1.4.1)

- **Question:** When I use Java to write my program under some specific OS, I always set some environment variables to control the application program. For example, CLASSPATH.
What is about other variables that I should be aware of? And how can I set them?

  Answer: Java defines own environment which is different from the default one provided by the underlying OS.

  You can define your own "environment" by using

  java -Dname1=myvalue1 -Dname2=myvalue2

- **Question:** Where can I find sample code for J2EE guidelines, patterns, and code for end-to-end Java applications?

  Answer: J2EE Patterns catalogue is set of practical solutions to recurring problems! J2EE Patterns not only help you build simpler systems that work, but they also help you build beautiful programs
  J2EE Patterns catalogue is here:
  http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html

- **Question:** What is date of Java birthday?

  Answer: May 23, 1995. John Gage, director of the Science Office for Sun Microsystems, and Marc Andreessen, cofounder and executive vice president at Netscape, stepped onto a stage and announced to the SunWorld audience that Java technology was real, it was official, and it was going to be incorporated into Netscape Navigator, the world's portal to the Internet.

  At that time, the entire Java technology team, not yet a division, numbered less than 30 people.

  More about early history of Java please read here:
  http://java.sun.com/features/1998/05/birthday.html

- **Question:** Why does not exist "JavaNotFoundException"?
When I deliver my Java program to customers often Java is not installed on theirs machines. Having such exception I could in better form (GUI popup, for example) inform my customers about the problem...

  Answer: Is not it obvious that to be able to find the absence of Java with a Java program you need the Java :-)

Java pitfalls: loop without braces.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Java allows loop without braces. Quite common mistake happens with such code:

```
while (condition)

   statement1; statement2
```

Many think that both statements will be in the body of while loop. Only statement1 is in the body.

Null body loops like this:

```
while (condition); //  infinite loop

   statement;
```

runs forever!!!

- **Good to know:** In releases prior to 1.4.2, it was possible to create an invalid URL instance such as the following: new URL ("http://www.foo.com:-1/index.html"). Specifying a port number of -1 is not valid. However, in previous releases the -1 was stripped off when sending such URLS with HTTP commands. In 1.4.2, the -1 is left in the URL and this may cause problems for some HTTP servers or proxies. Applications should not create URL instances with a -1 port number. Assuming the -1 refers to the default port in this example, the correct way to create this URL is new URL ("http://www.foo.com/index.html"). This note only applies to the single argument URL constructor.

- **Question:** I was testing the new J2RE 1.4.2 for compatibility with our programs (written on Java 1.2.2 version) and found that the Installation of the J2RE on Windows has changed, with impact on the supported locales...

  Answer: Yes, you are right. this is written in release notes for J2RE 1.4.2:
  There now is a single installer, which by default installs a runtime with support for European languages if the Windows host system only supports such languages, or a runtime with support for all languages if the Windows host system supports at least one non-European language. Users can request installation of additional languages in a custom setup. For complete information on which locales and encodings are supported with which installation, see:

  the Supported Locales
  http://java.sun.com/j2se/1.4.2/docs/guide/intl/locale.doc.html

  and Supported Encodings documents
  http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html

- **Question:** Why SUN does not introduce dynamic policies (for security)?

  Answer: Support for dynamic policies has been added. In Java 2 SDK releases prior to version 1.4, classes were statically bound with permissions by querying security policy during class loading. The lifetime of this binding was scoped by the lifetime of the class loader. In version 1.4 this binding is now deferred until needed by a

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

security check. The lifetime of the binding is now scoped by the lifetime of the security policy.

- **Question:** Can I use system provided random numbers generators when I run my Java program on Windows? Not that that provided by SUN?

  Answer: The Sun SecureRandom implementation now also makes use of an operating system-provided entropy source on Windows platforms, which can improve the start-up time of cryptographic applications considerably. Edit the <java.home>/lib/security/java.security to control this feature.

- **Question:** What are enhancements were added to new javac since 1.4.2?

  Answer: The following enhancements and bug fixes have been made to the javac source compiler in the Java 2 SDK.
  RFE 4626998 implemented: Added -cp synonym for -classpath option to set classpath.
  RFE 4515858 implemented: Added -Xbootclasspath/p: option to prepend to bootclasspath.
  RFE 4253402 implemented: Added -Xmaxerrors and -Xmaxwarns to increase the number of errors and warnings printed by javac.

  Also optimizations have been made to the core libraries to reduce start-up time. For a small command line application, start-up time has been reduced by roughly thirty percent; for a small Swing application, by fifteen to twenty percent.
  In some rare situations, one optimization related to filename canonicalization may cause inconsistent views of the host platform's file system to be briefly visible to Java applications. The optimization caches, for a short time, the results of calls to File.getCanonicalFile() and File.getCanonicalPath(). If a file is moved by another application then the canonicalization result for that file may briefly differ from its true value, although attempts to open and access the file will succeed or fail as before. Applications should not rely on the presence or absence of such inconsistencies. If an application must disable this optimization for correctness purposes then the system property -Dsun.io.useCanonCaches=false may be specified.

- **Good to know:** Performance testing has shown that the finalize methods in the channel classes impose a significant GC penalty upon server-type programs that must deal with thousands of new open channels per second.

  And now in J2SE 1.4.2the finalize methods of the primitive channel classes (SocketChannel, ServerSocketChannel, etc.) have been removed . Additional performance test measurements revealed that the presence of these methods imposed a significant (factor of 100) garbage-collection penalty upon server-type applications that process thousands of new open channels per second. User code that deals in channels should be carefully written to close channels explicitly rather than rely upon the garbage collector to do so.

- **Question:** If I turn off daylight saving during an installation of Windows XP then Java runtime detects the platform time zone in the GMT offset format (e.g., "GMT+09:00"), not as a time zone ID (e.g., "Asia/Tokyo"). What is fix for that problem?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: To fix this installation problem, take the following steps after the installation (unless you intend to turn off the setting):
1. Open Date/Time in Control Panel.
2. On the Time Zone tab, choose a time zone that observes daylight saving time (e.g., "(GMT-08:00) Pacific Time (US & Canada); Tijuana"), select the "Automatically adjust clock for daylight saving changes" check box, and press the Apply button.
3. Choose your time zone back and press the OK button.

- **Question:** I am getting strange artefacts when rendering Asian text in a TextArea component. I run Japanese version of Windows NT...

Answer: This problem occurs if you install English version of SW onto non-Latin versions of NT. For example, Visual C++ or Office 97.
The problem is caused when the installation of those programs replaces the Asian DLL Riched32.dll file with the English version of the same file. The problem can be corrected by replacing Riched32.dll with the Asian version.

- **Question:** If I run my applet with the SUN VM I get exception: "Either "code" or "object" should be specified, but not both." It does not happen with the Microsoft VM. Part 1

Answer: I found good recommendations on Sun's website about how to install latest browsers and avoid possible problems with running applets in different browsers:

"This is a compatibility issue. To avoid the exception with the Sun VM, do not use an attribute named object with the PARAM element in an applet.
For Netscape 7, or Mozilla 1.1 or later, running on Microsoft Windows, the browser will always use Sun's latest JRE—even if you have deselected the "Netscape 6 or later" or "Mozilla 1.1 or later" check box in the Browser tab of the Java Plug-in Control Panel. This is due to the auto-scan features of these browsers. For instance, if Netscape 7 is installed on a Windows machine that has Sun's 1.4.2 JRE installed as the latest version and you deselect "Netscape 6 or later" in the Control Panel, Netscape 7 will still run Sun's 1.4.2 JRE (Java Plug-in).

Using Java Plug-in 1.4.2 in conjunction with Netscape TM 6.x browsers can cause erratic behavior, including closure of the browser window, for applets that invoke applet functions from JavaScript. This problem is due to a bug in the browser, and you can avoid this problem by using Netscape 7.0 browsers.

- **Question:** If I run my applet with the SUN VM I get exception: "Either "code" or "object" should be specified, but not both." It does not happen with the Microsoft VM. Part 2

The Netscape 7 installer provides three choices for installation: Recommended, Full, and Custom. The Java Plug-in may not work properly in Recommended or Full installations of Netscape 7 browsers. Such Netscape 7 browsers may close when attempting to launch an applet. To avoid this problem, install Netscape 7 using the Custom option, and choose not to install the Quality Feedback Agent.

When installing Java Plug-in, it is necessary to shut down and then restart your web browser. On Microsoft Windows platforms, Netscape 7 has a Quick Launch feature that keeps it running in the background so that it can be launched with minimal start-up time from a Quick Launch icon in the system tray. To completely shut down

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Netscape 7, you must exit Netscape 7 from its Quick Launch icon in the system tray. "

This tip is based on:
http://java.sun.com/j2se/1.4.2/relnotes.html#vm

- **Good to know:** If your system does take in external XML data this security tip is for you!

While XML does not allow recursive entity definitions, it does permit nested entity definitions, which produces the potential for Denial of Service attacks on a server which accepts XML data from external sources. For example, a SOAP document like the following that has extremely deeply nested entity definitions can consume 100% of CPU time and a lot of memory in entity expansions.

```
<?xml version="1.0" encoding ="UTF-8"?>
<!DOCTYPE foobar[
<!ENTITY x100 "foobar">
<!ENTITY x99 "&x100;&x100;">
<!ENTITY x98 "&x99;&x99;">
...
<!ENTITY x2 "&x3;&x3;">
<!ENTITY x1 "&x2;&x2;">
]>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=...>
<SOAP-ENV:Body>
<ns1:aaa xmlns:ns1="urn:aaa" SOAP-ENV:encodingStyle="...">
<foobar xsi:type="xsd:string">&x1;</foobar>
</ns1:aaa>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

A system that doesn't take in external XML data need not be concerned with issue, but one that does can utilize one of the following safeguards to prevent the problem:
New system property to limit entity expansion
The entityExpansionLimit system property lets existing applications constrain the total number of entity expansions without recompiling the code. The parser throws a fatal error once it has reached the entity expansion limit. (By default, no limit is set, because such a constraint would make the XML parser incompatible with the XML 1.0 specification.)

To set the entity expansion limit using the system property, use an option like the following on the java command line: -DentityExpansionLimit=100000

New parser property to disallow DTDs
The application can also set the http://apache.org/xml/features/disallow-doctype-decl parser property to true. A fatal error is then thrown if the incoming XML document contains a DOCTYPE declaration. (The default value for this property is false.) This property is typically useful for SOAP based applications where a SOAP message must not contain a Document Type Declaration.

This tip is based on:
http://java.sun.com/j2se/1.4.2/relnotes.html#jaxp

- **Question:** In Ukraine first day of week is Monday but not Sunday

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

By defaults in the calendar object obtained with locale ua_UA getFirstDayOfWeek () function return SUNDAY as a first week day... Java version is 1.3.1_01

Answer: Yes indeed, according to this list:
http://www.pjh2.de/datetime/weeknumber/wnd.php
the first day in Ukraine is Monday. For those who does not know will say that in many countries Sunday or even Saturday is first day, for example in Egypt - Saturday and in Argentina and Brasil - Sunday...
It was a bug since according to ISO 8601 in all European countries a week starts with Monday...
I know that with Hungarian calendar was the same bug... In latest 1.4.1 and beyond the bug is fixed.

- **Question:** We compile a large number of files and we have strange problem:
javac -deprecation doesn't print a warning of a usage in file A.java of a deprecated method in B.java if B.java is being
compiled at the same time as A.java.
It is big problem since we want to know if we use deprecated code...

Answer: Yes the problem presents in all Java versions since 1.1.7 and then in 1.2, 1.3 and 1.4
You can compile file by file, although it is not convenient :-(
I hope this will be corrected soon by Sun!

- **Question:** How can I completely clean up my Windows XP from Microsoft JVM before I install Sun Microsystems JVM?

Answer: You might want to remove the Microsoft JVM, which Microsoft no longer supports, in favour of the more recent Sun Microsystems JVM. To remove the Microsoft JVM, perform the following steps:
1. From the Start menu, select Run.
2. Enter the command
RunDll32 advpack.dll,LaunchINFSection java.inf,UnInstall
to start the uninstall process
3. Click Yes to the confirmation, then select Reboot.
4. After the machine restarts, delete the following items:
the \%systemroot%\java folder
java.pnf from the \%systemroot%\inf folder
jview.exe and wjview.exe from the \%systemroot%\system32 folder
The HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Java VM registry subkey
The HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\AdvancedOptions\JAVA_VM registry subkey (to remove the Microsoft Internet Explorer (IE) options)

Now you can get latest JVM from Sun!

- **Question:** When does garbage collection performance matter to the user?

Answer: For many applications it doesn't. That is, the application can perform within its specifications in the presence of garbage collection with pauses of modest frequency and duration. An example where this is not the case (when the default collector is used) would be a large application that scales well to large number of threads, processors, sockets, and a large amount of memory.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Question: Can my program interact with Garbage Collector and get notified about a status of object (going to be GCollected, already collected)?

Answer: You should use java.lang.ref package!

This package provides reference-object classes, which support a limited degree of interaction with the garbage collector. A program may use a reference object to maintain a reference to some other object in such a way that the latter object may still be reclaimed by the collector. A program may also arrange to be notified some time after the collector has determined that the reachability of a given object has changed.

Going from strongest to weakest, the different levels of reachability reflect the life cycle of an object. They are operationally defined as follows:

> An object is strongly reachable if it can be reached by some thread without traversing any reference objects. A newly-created object is strongly reachable by the thread that created it.
>
> An object is softly reachable if it is not strongly reachable but can be reached by traversing a soft reference.
>
> An object is weakly reachable if it is neither strongly nor softly reachable but can be reached by traversing a weak reference. When the weak references to a weakly-reachable object are cleared, the object becomes eligible for finalization.
>
> An object is phantom reachable if it is neither strongly, softly, nor weakly reachable, it has been finalized, and some phantom reference refers to it.
>
> Finally, an object is unreachable, and therefore eligible for reclamation, when it is not reachable in any of the above ways.

Based on Java API description

- **Question:** What is young generation in GC (Garbage Collector)?

Answer: In the J2SE platform version 1.4.1 two new garbage collectors were introduced to make a total of four garbage collectors from which to choose.

Beginning with the J2SE platform, version 1.2, the virtual machine incorporated a number of different garbage collection algorithms that are combined using generational collection. While naive garbage collection examines every live object in the heap, generational collection exploits several empirically observed properties of most applications to avoid extra work.

The default collector in HotSpot has two generations: the young generation and the tenured generation. Most allocations are done in the young generation. The young generation is optimized for objects that have a short lifetime relative to the interval between collections. Objects that survive several collections in the young generation are moved to the tenured generation. The young generation is typically smaller and is collected more often. The tenured generation is typically larger and collected less often.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

The young generation collector is a copying collector. The young generation is divided into 3 spaces: eden-space, to-space, and from-space. Allocations are done from eden-space and from-space. When those are full a young generation is collection is done. The expectation is that most of the objects are garbage and any surviving objects can be copied to to-space. If there are more surviving objects than can fit into to-space, the remaining objects are copied into the tenured generation. There is an option to collect the young generation in parallel.

The tenured generation is collected with a mark-sweep-compact collection. There is an option to collect the tenured generation concurrently.

This tip is based on FAQ about Garbage Collection in the HotSpot Java Virtual Machine
http://java.sun.com/docs/hotspot/gc1.4.2/faq.html
and article "Tuning Garbage Collection with the 1.4.2 Java Virtual Machine"
http://java.sun.com/docs/hotspot/gc1.4.2/

- **Question:** What are additional types of GC do exist in the J2SE platform, version 1.4.2?

Answer:

The throughput collector: this collector uses a parallel version of the young generation collector. It is used if the -XX:+UseParallelGC option is passed on the command line. The tenured generation collector is the same as the default collector.
The concurrent low pause collector: this collector is used if the -XX:+UseConcMarkSweepGC is passed on the command line. The concurrent collector is used to collect the tenured generation and does most of the collection concurrently with the execution of the application. The application is paused for short periods during the collection. A parallel version of the young generation copying collector is used with the concurrent collector if the combination -XX:+UseConcMarkSweepGC -XX:+UseParNewGC is passed on the command line.
This kind is also known as concurrent GC.
The incremental (sometimes called train) low pause collector: this collector is used only if -Xincgc is passed on the command line. By careful bookkeeping, the incremental garbage collector collects just a portion of the tenured generation at each minor collection, trying to spread the large pause of a major collection over many minor collections. However, it is even slower than the default tenured generation collector when considering overall throughput.

Important! -XX:+UseParallelGC should not be used with -XX:+UseConcMarkSweepGC. The argument parsing in the J2SE platform, version 1.4.2 should only allow legal combinations of command line options for garbage collectors, but earlier releases may not detect all illegal combinations and the results for illegal combinations are unpredictable.

Also important to remember: always try default GC before using additional GCs!

- **Question:** What type of garbage collection does a System.gc() do?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: An explicit request to do a garbage collection does a full collection (both young generation and tenured generation). A full collection is always done with the application paused for the duration of the collection.

- **Question:** My application runs much slower with 1.3/1.4 when compared with the 1.2 Production release for Solaris. Why?

Answer: Assuming that you are running a lot of bytecode make sure that you are using the correct mode of the virtual machine. For applications where small footprint and fast start up are important, use -client. For applications where overall performance is the most important, use -server. Don't forget that -server or -client must be the first argument to java, -client is the default. If this isn't your problem, read on for more tuning parameters you can try...

from the Java Hotspot Virtual Machine FAQ:
http://java.sun.com/docs/hotspot/PerformanceFAQ.html

- **Question:** What's the difference between the -client and -server systems (options in JVM)?

Answer: These two systems are different binaries. They are essentially two different compilers (JITs) interfacing to the same runtime system. The client system is optimal for applications which need fast start-up times or small footprints, the server system is optimal for applications where the performance is most important. In general the client system is better on GUIs. Some of the other differences include the compilation policy used, heap defaults, and inlining policy.

from the Java Hotspot Virtual Machine FAQ:
http://java.sun.com/docs/hotspot/PerformanceFAQ.html

- **Question:** With 1.3.1 we got 4GB heaps on Solaris, why can't I get this to work on Windows?

Answer: It's due to fragmentation of the address space. After Windows has loaded all its stuff, plus the java dlls, the largest available continuous chunk of addresses for the heap is about 1.4-1.6G on Win/NT. Might be smaller on Win/98.

If you really need more space they can try rebasing java.exe and the JDK dlls higher. But that won't buy you much (maybe a couple hundred meg).
from the Java Hotspot Virtual Machine FAQ:
http://java.sun.com/docs/hotspot/PerformanceFAQ.html

- **Question:** Is it possible to use RSS ("Really Simple Syndication") together with JSP?

Answer: Yes, SUN recently developed RSS Utilities package - a JSP Tag Library to be used by anybody with a basic understanding of RSS, JavaServer Pages, and HTML. The taglib is mostly geared towards non-technical editors of web sites that use RSS for aggregating news content.
It contains a set of custom JSP tags which make up the RSS Utilities Tag library, and a flexible RSS Parser.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Please read more here:
http://developer.java.sun.com/developer/technicalArticles/javaserverpages/rss_utilities/

- **Question:** I use an old Java Tool from SUN. Today I read that SUN started End Of Life Process for this Java tool.
What does it mean? Does it mean that when the process will be finished my tool will stop to work? What is about customers?

  Answer: Everything is Ok! Life continues and the tool will work :-)
During the EOL transition period, all announced products will be supported as per existing customer support agreements. After the EOL transition period, these products will no longer be supported by Sun. Products that have completed the EOL process will be moved to the Archive area here:
http://java.sun.com/products/archive/index.html

- **Question:** I use the MediaTracker class is a utility class to track the status of a number of media objects. I need to load images over the net. How could I check the progress and know when loading is finished?

  Answer: Use an asynchronous update interface for receiving notifications about Image information as the Image is constructed - ImageObserver. The imageUpdate method gets called with infoflag 8 (SOMEBITS) every time when more pixels arrive. And with infoflag 32 (ALLBITS) when loading is complete.

  --
  alex

- **Question:** ArrayList vs. LinkedList

  Answer: The List interface defines an ordered collection of elements. Available implementations include ArrayList and LinkedList, where both implement the List interface. When your program needs to work with a List, it should not care if it is an ArrayList or a LinkedList, only that whatever class is used provides a known behavior. That behavior is the interface.

  If your program frequently provides random access to the data of the list, the ArrayList class offers quick access to individual elements of the list. This quick access comes at a cost of slower operations for adding and removing in the middle of the list.
If this latter behavior is what you desire, than the LinkedList class offers a better alternative. It provides quick sequential access, additions, and deletes, at a cost of slower random access.

  The tip based on Sun's newsletter

- **Question:** How does JVM interact with profilers?

  Answer: The Java Virtual Machine has Profiler Interface (JVMPI) in the Java 2 SDK, Standard Edition. This interface is an experimental feature in the Java 2 SDK. The JVMPI is not yet a standard profiling interface. It is intended for tools vendors to develop profilers that work in conjunction with Sun's Java virtual machine implementation.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

"The JVMPI is a two-way function call interface between the Java virtual machine and an in-process profiler agent. On one hand, the virtual machine notifies the profiler agent of various events, corresponding to, for example, heap allocation, thread start, etc. On the other hand, the profiler agent issues controls and requests for more information through the JVMPI. For example, the profiler agent can turn on/off a specific event notification, based on the needs of the profiler front-end.

The profiler front-end may or may not run in the same process as the profiler agent. It may reside in a different process on the same machine, or on a remote machine connected via the network. The JVMPI does not specify a standard wire protocol. Tools vendors may design wire protocols suitable for the needs of different profiler front-ends.

A profiling tool based on JVMPI can obtain a variety of information such as heavy memory allocation sites, CPU usage hot-spots, unnecessary object retention, and monitor contention, for a comprehensive performance analysis.
JVMPI supports partial profiling, i.e. a user can selectively profile an application for certain subsets of the time the virtual machine is up and can also choose to obtain only certain types of profiling information.

In the current version of JVMPI, only one agent per virtual machine can be supported." Details can be found here:
http://java.sun.com/j2se/1.4.1/docs/guide/jvmpi/jvmpi.html

- **Question:** Could you tell me about chain exceptions?

Answer: It is common for Java code to catch one exception and throw another:

```
try {
...
} catch(YourException e) {
   throw new MyException();
}
```

Unfortunately, the information contained in the "causative exception" (YourException in the example above) is generally lost, which greatly complicates debugging.

The version 1.4 libraries add a unified, general-purpose mechanism to record such information. The pre-existing ad-hoc facilities have been retrofitted to use the new technique. Two new constructors were added to Throwable. Each takes another Throwable as the cause.

```
Throwable(Throwable cause)
Throwable(String message, Throwable cause)
```

Common exceptions classes have had analogous new constructors added. The intended usage is:

```
try {
   lowLevelOp();
} catch (LowLevelException le) {
   throw new HighLevelException(le);
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Legacy exception classes can use a modified form of new idiom through the Throwable initCause(Throwable cause) method, such as:

```
try {
    lowLevelOp();
} catch (LowLevelException le) {
    throw new (HighLevelException)
    HighLevelException().initCause(le);
}
```

The cause information can be retrieved through the method Throwable getCause(). For more information, see:
http://java.sun.com/j2se/1.4/docs/guide/lang/chainedexceptions.html

- **Question:** How can my application know how many CPUs work in a system?

  Answer: Use 1.4.x JDK... The method activeProcessorCount() was added to the java.lang.Runtime class.
  Sun Microsystems, Inc. 12 of 37
  The activeProcessorCount() method returns the number of hardware processors currently available to the VM. Large applications will typically use
  this value to calculate the number of worker threads they should create. On some operating systems, such as the Solaris operating environment (OE) and Microsoft Windows NT and 2000, the processor count may change dynamically.

  An application will, in general, have to poll to determine when the count changes.

  The tip is based on Sun's "Version 1.4 Core Libraries" description

- **Question:** I read that in the Java (version 1.4.x) reflection implementation has been optimized for speed and it is significantly faster now...?

  Answer: The APIs and semantics are the same as in previous J2SE versions. There are three key optimizations in the new implementation:

  Caching of results of Class.getFields(), Class.getMethods(), and related routines
  Dynamic generation of bytecodes for Method.invoke() and

  Constructor.newInstance()

  Optimized security checks for reflective calls

  Due to these optimizations, reflective method calls are between 20 and 25 times faster, and reflective object creation is between 5 and 6 times faster. Reflection intensive code, such as serialization, has benefited from this additional speed.

  The tip is based on Sun's "Version 1.4 Core Libraries" description

- **Good to know:** JFluid - a profiling tool for the Java programming language that allows you to profile an arbitrary subset of your program, that can be changed on-the-fly, while the program is running. This is a capability not available in any other

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

profiling tool for Java at this time. Since JFluid relies on our unique dynamic bytecode instrumentation mechanism, it can run only on our specially modified HotSpot VM (currently derived from JDK(tm) version 1.4.2).

In JFluid, you can select your application's methods to profile by pointing at a single method. This method is then treated as a call subgraph root. JFluid instruments this method, and also all methods that it calls, transitively. In other words, it dynamically reveals a whole call subgraph. Special bytecode packets are injected into these methods on-the-fly, so that they start to generate profiling events such as method entry and exit. These events are recorded, and then the obtained performance data is displayed in the standard forms of context call tree (what you see e.g. in OptimizeIt) and sorted accumulated method time list.

Alternatively, you can instrument and profile an arbitrary code fragment within a method. This kind of instrumentation is extremely lightweight and inobtrusive, therefore in this way you can obtain very precise CPU time measurements.

Finally, you can activate and deactivate memory profiling that allows you to see the number and types of all objects allocated by the program, the locations where these objects are allocated, and the call paths at the moment of allocation. We are using a special semi-statistical technique for gathering information about call paths that results in a highly controllable, and usually quite modest, overhead when collecting this kind of information.

To learn more about the JFluid tool, read the documentation page (http://research.sun.com/projects/jfluid/help.html) (this is the same page that is available from within the tool when you hit "Help"). Click here (http://research.sun.com/projects/jfluid/download/) to download JFluid.

Future:
At present JFluid is an experimental technology. The possibility of productizing the JVM changes necessary to run JFluid, as well as the tool itself, is being discussed, but no firm decisions have been taken yet. Note that if the JVM API that is currently used by JFluid becomes a standard, it could then be used by any tool from any vendor, for example for inobtrusive EJB application monitoring, fault injection, and other interesting things.
All info is here:
http://research.sun.com/projects/jfluid/

- **Question:** What is Signal Chaining in JVM?

Answer: The Java TM 2 SDK, Standard Edition, (J2SDK) v1.4 contains a new signal-chaining facility. Signal-chaining enables the Java Platform to better interoperate with native code that installs its own signal handlers. The facility works on both Solaris and Linux platforms.

The signal-chaining facility was introduced to remedy a problem with signal handling in previous versions of the Java HotSpot VM. Prior to version 1.4, the Java HotSpot VM would not allow application-installed signal handlers for certain signals including, for example, SIGBUS, SIGSEGV, SIGILL, etc, since those signal handlers could conflict with the signal handlers used internally by the Java HotSpot VM.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

The signal-chaining facility offers:

A. Support for pre-installed signal handlers when the HotSpot VM is created.
B. Support for signal handler installation after the HotSpot VM is created, inside JNI code or from another native thread.

Pre-installed signal handlers (A) are supported by means of saving existing signal handlers, for signals that are used by the VM, when the VM is first created. Later, when any of these signals are raised and found not to be targeted at the Java HotSpot VM, the pre-installed handlers are invoked. In other words, pre-installed handlers are "chained" behind the VM handlers for these signals.

more please read here:
http://java.sun.com/j2se/1.4.1/docs/guide/vm/signal-chaining.html

- **Question:** How can I use the new error-reporting mechanism in JVM?

Answer: When the Java Virtual Machine (JVM) detects a crash in native code such as JNI code written by the developer or when the JVM itself crashes, it will print debug information about the crash. This error message normally will include information such as the function name, library name, source-file name and line number at which the error occurred. (Currently, information about file name and line number are available only available on Microsoft Windows platforms.) For an example of a message emitted by the error handler, see JNI Error example.
The information provided by the new error-reporting mechanism will allow developers to more easily and efficiently debug their applications. If an error message indicates a problem in the JVM code itself, it will allow a developer to submit a more accurate and helpful bug report.

Sometimes the error-reporting mechanism won't be able to determine the information that might be helpful in locating the source of a crash. To get the most out of the error handler, developers should be aware of the following guidelines and limitations.

Compile in debug mode

In some circumstances, the error-reporting mechanism won't be able to determine symbol names. The most common reason for this is that the binary code that crashed was not compiled in debug mode and therefore doesn't have symbol tables. Developers should compile their code in debug mode to ensure that it contains the necessary debug information. In Visual Studio, for example, this means choosing "Debug" rather than "Release" as the project's build mode. When using gcc or cc on Linux or on the Solaris TM operating environment, compile using the -g command-line option.

More limitations are described in the article here:
http://java.sun.com/j2se/1.4.1/docs/guide/vm/error-handling.html

- **Question:** I understand that the XML is some kind of language to describe documents. Why do we need the XML for data exchange between systems?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: XML is used to aid the exchange of data. It makes it possible to define data in a clear way.
Both the sending and the receiving party will use XML to understand the kind of data that's been sent. By using XML everybody knows that the same interpretation of the data is used.

XML makes communication easy. It's a great tool for transactions between businesses.
But it has much more possibilities.
You can define other languages with XML. A good example is WML (Wireless Markup Language), the language used in WAP-communications.
WML is just an XML dialect.

Web services depend on the ability of parties to communicate with each other even if they are using different information systems. XML (Extensible Markup Language), a markup language that makes data portable, is a key technology in addressing this need. Enterprises have discovered the benefits of using XML for the integration of data both internally for sharing legacy data among departments and externally for sharing data with other enterprises. As a result, XML is increasingly being used for enterprise integration applications, both in tightly coupled and loosely coupled systems. Because of this data integration ability, XML has become the underpinning for Web-related computing.

- **Question:** What is an XML registry and JAXR?

Answer: XML registries are a vital component in the implementation of shared data exchanges. Developers looking to express information using XML need support in establishing common lexicons and grammars. This Registry enables the consistent use of XML, both vertically within projects and horizontally across organizations.

JAXR (Java API for XML Registries) enables Java software programmers to use a single, easy-to-use abstraction API to access a variety of XML registries. A unified JAXR information model describes content and metadata within XML registries.

JAXR gives developers the ability to write registry client programs that are portable across different target registries. JAXR also enables value-added capabilities beyond those of the underlying registries.

The current version of the JAXR specification includes detailed bindings between the JAXR information model and both the ebXML Registry and the UDDI version 2 specifications. You can find the latest version of the specification at:
http://java.sun.com/xml/downloads/jaxr.html

- **Question:** What is meant by "transformation" in JAXP?

Answer: The Extensible Stylesheet Language (XSL) includes both a transformation language and a formatting language. Each of these, naturally enough, is an XML application. The transformation language provides elements that define rules for how one XML document is transformed into another XML document. The transformed XML document may use the markup and DTD of the original document or it may use a completely different set of elements. In particular, it may use the elements defined by the second part of XSL, the formatting objects.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** How do I use JavaBeans components (beans) from a JSP page?

Answer: The JSP specification includes standard tags for bean use and manipulation. The useBean tag creates an instance of a specific JavaBeans class. If the instance already exists, it is retrieved. Otherwise, it is created. The setProperty and getProperty tags let you manipulate properties of the given object. These tags are described in more detail in the JSP specification and tutorial.
tip is from the JSP FAQ: http://java.sun.com/products/jsp/faq.html

- **Question:** What is Model-View-Controller (MVC) architecture?

Answer: The MVC paradigm (design pattern) is a way of breaking an application, or even just a piece of an application's interface, into three parts: the model, the view, and the controller. MVC was originally developed to map the traditional input, processing, output roles into the GUI realm:

Input --> Processing --> Output
Controller --> Model --> View

The user input, the modelling of the external world, and the visual feedback to the user are separated and handled by model, viewport and controller objects. The controller interprets mouse and keyboard inputs from the user and maps these user actions into commands that are sent to the model and/or viewport to effect the appropriate change. The model manages one or more data elements, responds to queries about its state, and responds to instructions to change state. The viewport manages a rectangular area of the display and is responsible for presenting data to the user through a combination of graphics and text.

Separating responsibilities among model, view, and controller objects reduces code duplication and makes applications easier to maintain. It also makes handling data easier, whether adding new data sources or changing data presentation, because business logic is kept separate from data. It is easier to support new client types, because it is not necessary to change the business logic with the addition of each new type of client.

Source: comp.object newsgroup and Sun's web site.

- **Question:** Where can I find brief description for common set of design patterns which I can use in my application (J2EE platform)? Part 1

Answer: look at this list below:

Intercepting filter--This pattern applies to request pre- and post-processing. It applies additional services needed to process a request. For example, an intercepting filter such as a servlet filter may handle all incoming requests to the Web site and provide a central mechanism for authorization.

View helper--A view helper encapsulates the presentation and data access logic portions of a view, thus refining the view and keeping it simpler. Presentation logic concerns formatting data for display on a page, while data access logic involves retrieving data. View helpers are often JSP tags for rendering or representing data and JavaBeans for retrieving data.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Composite view--This pattern makes view presentation more manageable by creating a template to handle common page elements for a view. Often, Web pages contain a combination of dynamic content and static elements, such as a header, footer, logo, background, and so forth. The dynamic portion is particular to a page, but the static elements are the same on every page. The composite view template captures the common features.

Front controller--This pattern provides a centralized controller for managing requests. A front controller receives all incoming client requests, forwards each request to an appropriate request handler, and presents an appropriate response to the client.

- **Question:** Where can I find brief description for common set of design patterns which I can use in my application (J2EE platform)? Part 2

Value object--This pattern facilitates data exchange between tiers (usually the Web and EJB tiers) by reducing the cost of distributed communication. In one remote call, a single value object can be used to retrieve a set of related data, which then is available locally to the client. See Chapter 5 for more information on value objects.

Session facade--This pattern coordinates operations between cooperating business objects, unifying application functions into a single, simplified interface for presentation to the calling code. It encapsulates and hides the complexity of classes that must cooperate in specific, possibly complex ways, and isolates its callers from business object implementation changes. A session facade, usually implemented as a session bean, hides the interactions of underlying enterprise beans.

Business delegate--This pattern intervenes between a remote business object and its client, adapting the business object's interface to a friendlier interface for the client. It decouples the Web tier presentation logic from the EJB tier by providing a facade or proxy to the EJB tier services. The delegate takes care of lower-level details, such as looking up remote objects and handling remote exceptions, and may perform performance optimizations, such as caching data retrieved from remote objects to reduce the number of remote calls.

Data access object--This pattern abstracts data access logic to specific resources. It separates the interfaces to systems resource from the underlying strategy used to access that resource. By encapsulating data access calls, data access objects facilitate adapting data access to different schemas or database types. See Chapters 5 and 6 for more information on data access objects.

I found this list which was published in "Designing Enterprise Applications with the J2EETM Platform, Second Edition" book and available online here: http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/DEA2eTOC.html

- **Question:** I am trying to make my code compact. I already "JARed" all files... Is it possible to make the JAR even smaller?

Answer: Yes, it is possible to make a JAR file smaller by using code obfuscators. Most of the modern obfuscators replacing long class/method, variables names (for example: MyFavoriteSistersBlueClass) by shorter, just 2-3 characters...
Use it on final step, before delivery to customer, not for further development, since

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

this procedure makes a code difficult to read and follow... In some cases a size can be reduced up to 30%!!!

- **Question:** Can you tell me about "100 Percent Pure Java Certification Program ". My project leader read somewhere that our SW should pass this certification. He also says that code for this program can be written only by Java certified programmers. I do not want to certify myself; I believe that I know already Java (working 3 years in the industry).

Answer: You can sleep easily, the Sun's "100 Percent Pure Java Certification Program " is over and is no longer supported by Sun. SUN used the program from the beginning to make the Java popular and clear (for example to keep away from Microsoft's influence).

- **Question:** Do you know any free Java editor for the Java?

Answer: jEdit! Giving you an overview I took the info from JEdit home page:
"Overview
jEdit is a mature and well-designed programmer's text editor that has been in development for over 5 years.

While jEdit beats many expensive development tools for features and ease of use, it is released as free software with full source code, provided under the terms of the GNU General Public License.

The core of jEdit is primarily developed by Slava Pestov, and jEdit plugins are being written by a large and diverse team of programmers from around the world.

Some of jEdit's features include:

- Written in Java, so it runs on MacOS X, OS/2, UNIX, VMS and Windows.

- Built-in macro language; extensible plugin architecture. Dozens of macros and plugins are available.

- Plugins can be downloaded and installed from within jEdit using the "plugin manager" feature.

- Auto indent, and syntax highlighting for more than 80 languages.

- Supports a large number of character encodings including UTF8 and Unicode.
- Folding for selectively hiding regions of text.

- Word wrap.

- Highly configurable and customizable.

Every other feature, both basic and advanced, you would expect to find in a text editor. See the Features page for a full list.

See all the features here: http://www.jedit.org/index.php?page=features "
Home page for the project is here: http://www.jedit.org/

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I have found that ZipEntry() method behaves strange: it does not convert filenames from Unicode to platform specifics

  Answer: yes, it is known Java bug and discovered in the next version of Java 1.2.1, 1.2.2, 1.3, 1.3.1, 1.4.0_01, 1.4.1..

- **Question:** In Socket class exists method "getOOBInline" that checks if OOBINLINE is enabled. What is "OOBINLINE" and can I do with it?

  Answer: OOB stands for Out-Of-Band Data.

  The TCP/IP protocol allows users to send data asynchronously, otherwise known as OUT-OF-BAND data. When using this feature, the operating system interrupts the receiving process if this process has chosen to be notified about out-of-band data. The receiver can grab this input without affecting any information currently queued on the socket.
  Therefore, you can use this without interfering with any current activity due to other wire and remote interfaces.

  Unfortunately, most implementations of TCP/IP are broken, so use of out-of-band data is limited for safety reasons. You can only reliably send one character at a time. What is about safety? Let's look at this scenario: While a server is working on a task, the server is sensitive to interrupts. An interrupt will cause the server to abort the current task and begin listening for new instructions.
  Interrupts can be sent to the server by sending an interrupt ("INT") to the process number of the server with the Unix kill command. The server socket will also raise an interrupt if out of band (OOB) data are received. Thus, the client can send a single arbitrary byte of OOB data to generate an interrupt. So, if someone will send much OOB data to a server it can seriously decrease performance of server...

  On one of hacker's web sites I found "OOB Attack Program" that described like this: "This is a good program that exploits the oob attack on port 139 of any Windows machine running MS TCP/IP. Allows for multiple IP addresses to be attacked. Coded for fun by: Hiro Protagonist "

  If you are developing a server that will be exposed to Internet remember that some people "for fun" can take it down because OOBINLINE is enabled.

- **Question:** I found that in Java API (when looked at v1.4.2) exist "PrivilegedExceptionAction" and "PrivilegedActionException". It takes a while to try to distinguish the meanings of two names. Why naming of classes is done so bad and confuses people. Is not it enough another words?

  No answer, just comment: I think it is good sign :-) Java became big enough and has almost all possible methods and classes. We feel the luck of words to describe a wealth of Java :-)

Question: I was using Java 1.4.x during of development of program. Our customers want to use this version of Java because it causes some compatibility troubles for them... What could I do? I tried to insist on changing of Java but it did not help :-(

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

- **Answer:** You can still use newer version of Java, but when you compile use the option "-target". For example:
  javac -target 1.3

- **Question:** Where could I buy gifts to my colleagues (t-shirts, glasses, caps, mugs and so on) with Java logo?

  Answer: One place which I know is Java certified :-) since the link I found on Sun's home page) here:
  http://www201.ikiosk.com/cgi-shl/index.cgi?shop=33293&date=2002%3A04%3A26%3A10%3A29%3A26

- **Question:** Where can I find commercial software that is written on Java? We need often some ready components for fast development and ready to pay for that!

  Answer: I know one site which is good resource who is looking for different programs and components (beans, applets, libraries) for your software development: http://www.javashareware.com

- **Question:** Could you point me Java Data Binding Tools (to generate Java code from instances of WXS in order to represent the structures defined therein)?

  Answer: Exists a few such tools. They generate Java object models from XML structures and encapsulates the low-level details of working with instances of XML documents and provide an "an alternative to the DOM" for processing XML from Java.

  Java XML Binders simplify your XML development by creating Java classes that encapsulate XML parsing and validation and which have methods that map directly to your XML data elements and attributes.

  Breeze XML Binder
  Breeze is a commercial tool produced by Breeze Factor. It's available as an evaluation copy from http://www.breezefactor.com/.
  If you want to use the software and the generate code you have to pay a license fee.

  Castor
  Castor is an open source data binding framework for Java[tm]. It's basically the shortest path between Java objects, XML documents and SQL tables. Castor provides Java to XML binding, Java to SQL persistence, and then some more.
  It is available free of charge in binary or source form at http://www.castor.org. The code generation tool has a command line interface and a programmatic interface.

  JAXB Reference Implementation
  JAXB is a standard mapping developed by Sun in cooperation with a number of partners. JAXB in itself only specifies the intended behavior of a data binding tool and is not a tool in itself. However, the mapping is accompanied by a reference implementation. The JAXB jars and the reference implementation are both part of the Sun Java Web Services Developer Pack 1.2, which is available for free at http://java.sun.com/xml/jaxb/ .

  XGen
  XGen is an open source tool produced by Commerce One. The tool has been

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

developed with the Castor code as a basis, but with a substantial number of changes in the mapping and the functionality. The tool is included in the Commerce One Conductor DocSOAP XML Developer's Kit available free of charge at http://www.commerceone.com/developers/docsoapxdk

In this tip were used descriptions from the article:
http://www.xml.com/pub/a/2003/09/03/binding.html

In the article you can find test results for the listed above tools please look at this test.

- **Question:** I often read that Java folk discuss Java products that are not on the Sun's web site yet. I mean different alpha, beta releases and so on...
Where they are taking all those versions of products from?

  Answer: Most of Java products (from SUN I mean) are accessible through so named "Early Access". It is free to all Java Developer Connection(JDC) members. You can also register there and contribute into bug hunting process :-)
The list of products that are "Early Accessible" here:
http://developer.java.sun.com/developer/earlyAccess/index.html

  When product is ready SUN moves into public "products" area... This page is very valuable resource, because you can plan your development in advance without awaiting of product to be finally released.
Use of future APIs today!

- **Question:** I just started in the big Java project and my task is to review Javadoc comments in the current code and estimate how much left to do...
Does exist an "easy" way to do that?

  Answer: yes, probably if you will try to use Sun Doc Check Doclet. Although it is experimental code developed originally for internal use at Sun and you will use at your own risk, I advise you to try.
More info from SUN: "The Sun TM Doc Check Doclet is an extension to the Javadoc tool. It runs on source code and reviews documentation comments, generating an HTML report that identifies empty comments and other omissions and irregularities in the documentation comments. The report displays the number of doc errors, categorizes problems according to severity, and suggests text to include. (Those suggestions can then be copied into the code and edited.) It can be instructed to ignore packages and classes using wildcard patterns. An included utility package contains functionality that scans a directory tree to generate a list of packages, as well as pattern-matching classes for filenames."
You can read more about the tool and download it here:
http://java.sun.com/j2se/javadoc/doccheck/docs/index.html

- **Question:** How could I check if a class is serializable?

  Answer: You can use the utility that comes with JDK - serialver.
You can use it in command line or GUI based form:
c:\> serialver -show

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I have used quite long a few serializable classes in my application. Now when I added new functionality to one of them I get InvalidClassException exception. Why my class is invalid? It compiles, no problem...

  Answer: All serializable classes have unique serialVersionUID. The serialVersionUID is computed using the signature of a stream of bytes that reflect the class definition. The National Institute of Standards and Technology (NIST) Secure Hash Algorithm (SHA-1) is used compute a signature for the stream. The first two 32-bit quantities are used to form a 64-bit hash. A java.lang.DataOutputStream is used to convert primitive data types to a sequence of bytes. The values input to the stream are defined by the Java Virtual Machine (VM) specification for classes.

  So, in your case you need manually run serialver to generate a serialVersionUID for the old classes and manually assign them to newer versions. Add the serialVersionUID to the class as a "static final long".

  You are not required to define the serialVersionUID field in every class, but in the cases like yours you may find it useful.

- **Question:** Can I use my own long value for serialVersionUID in my class?

  Answer: If you don't have any previously serialized objects of that class around, you can use any serialVersionUID as you want.

  But in case if you have already older versions of your classes and want to keep compatibility you must use serialVersionUID which is obtained from the serialver tool.

- **Question:** What I should think about when I want secure serialization?

  Answer: If you develop some class that provides access to file system, for example, you must know that a file descriptor contains a handle that provides access to an operating system resource. Being able to forge a file descriptor would allow some forms of illegal access, since restoring state is done from a stream.

  Easiest way is to use static and static transient fields that contain sensitive information. Transient and static fields are not serialized or deserialized. Marking the field will prevent the state from appearing in the stream and from being restored during deserialization. Since writing and reading (of private fields) cannot be superseded outside of the class, the class's transient fields are safe.

- **Question:** I evolve my serialized class and do some changes to it. What are changes make next versions of my class incompatible (compatible)?

  Part 1.

  Answer: Actually it could not be better described (from my point of view) as it done in "Object Serialization Specification" by SUN.

  Incompatible Changes

  Incompatible changes to classes are those changes for which the guarantee of

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

interoperability cannot be maintained. The incompatible changes that may occur while evolving a class are:

- Deleting fields - If a field is deleted in a class, the stream written will not contain its value. When the stream is read by an earlier class, the value of the field will be set to the default value because no value is available in the stream. However, this default value may adversely impair the ability of the earlier version to fulfil its contract.

  Moving classes up or down the hierarchy - this cannot be allowed since the data in the stream appears in the wrong sequence.

  Changing a non-static field to static or a non-transient field to transient - This is equivalent to deleting a field from the class. This version of the class will not write that data to the stream so it will not be available to be read by earlier versions of the class. As in the deleting a field case above, the field of the earlier version will be initialized to the default value but that can cause the class to fail in unexpected ways.

  Changing the declared type of a primitive field - Each version of the class writes the data with its declared type. Earlier versions of the class attempting to read the field will fail because the type of the data in the stream does not match the type of the field.

- **Question:** I evolve my serialized class and do some changes to it. What are changes make next versions of my class incompatible (compatible)?

  Part 2.

  Incompatible Changes (continue)
  Changing the writeObject or readObject method so that it no longer writes or reads the default field data or changing it so that it attempts to write it or read it when the previous version did not. The default field data must consistently either appear or not appear in the stream.

  Changing a class from Serializable to Externalizable or visa-versa is an incompatible change since the stream will contain data that is incompatible with the implementation in the available class.

  Compatible Changes
  The compatible changes to a class are handled as follows:

- Adding fields - When the class being reconstituted has a field that does not occur in the stream, that field in the object will be initialized to the default value for its type. If class specific initialization is needed the class may provide a readObject method that can initialize the field to non-default values.

  Adding classes - The stream will contain the type hierarchy of each object in the stream. Comparing this hierarchy in the stream with the current class can detect additional classes. Since there is no information in the stream from which to initialize the object, the class's fields will be initialized to the default values.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Removing classes - Comparing the class hierarchy in the stream with that of the current class can detect that a class has been deleted. In this case, the fields and objects corresponding to that class are read from the stream. Primitive fields are discarded but the objects referenced by the deleted class are created since they may be referred to later in the stream. They will be garbage collected when the stream is garbage collected or reset.

Adding writeObject/readObject methods - If the version reading the stream has these methods then readObject is expected, as usual, to read the required data written to the stream by the default serialization. It should call defaultReadObject first before reading any optional data. The writeObject method is expected as usual to call defaultWriteObject to write the required data and then may write optional data.

- **Question:** I evolve my serialized class and do some changes to it. What are changes make next versions of my class incompatible (compatible)?

Part 3.

Compatible Changes (continue)

-

Removing writeObject/readObject methods - If the class reading the stream does not have these methods, the required data will be read by default serialization and the optional data will be discarded.

Adding java.io.Serializable - This is equivalent to adding types. There will be no values in the stream for this class so is fields will be initialized to default values. The support for subclassing non-serializable classes requires that the class's supertype have a no-arg constructor and the class itself will be initialized to default values. If the no-arg constructor is not available the NotSerializableException is thrown.

Removing java.io.Serializable so that it is no longer Serializable - This is equivalent to removing the class, and it can be dealt with by reading and discarding data for the class.

Changing the access to a field - The access modifiers public, package, protected and private have no effect on the ability of serialization to assign values to the fields.

Changing a field from static to non-static or transient to non-transient - This is equivalent to adding a field to the class. The new field will be written to the stream but earlier classes will ignore the value since serialization will not assign values to static or transient fields.

http://java.sun.com/products/jdk/rmi/doc/serial-spec/version.doc.html#6519

- **Question:** Is it possible with Java convert HTML documents to PDF ones?

Answer: Yes, with Sun MIF Doclet. The MIF Doclet is experimental code developed for internal use at Sun and released to the developer community for use as-is. It is not a supported product. Use at your own risk. While it was designed to be useful, bugs might not be fixed and features might not be added.

What is the MIF Doclet?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The MIF Doclet is an extension to the Javadoc tool that generates API documentation in MIF (Maker Interchange Format) from declarations and doc comments in source code in the Java TM programming language. It uses exactly the same set of Javadoc tags (such as @return, @param and @see) that the standard Doclet uses. FrameMaker batch processing programs enable these MIF documents to automatically be opened in Adobe FrameMaker and converted to FrameMaker native binary format, PDF or PostScript. Alternatively, you can manually choose "Save As" to save to Microsoft RTF, Microsoft Word or WordPerfect.

The goal has been to create a general, flexible means of conversion from doc comments to MIF that can be easily adapted to your own FrameMaker layout, style definitions and book organization. It tries to replicate much of the content generated by the standard HTML Doclet. However, the page layout has been modified from the standard HTML Doclet to add more layout options and to reduce vertical space, as page count needs to be conserved in print documentation. To obtain denser content, we put subheadings and data on the same line (rather than on separate lines) where possible, and eliminated repetition, such as by removing the "first sentence" from the index. The features list below covers the myriad options.

Take it from here:
http://java.sun.com/j2se/javadoc/mifdoclet/docs/mifdoclet.html#batchprograms

- **Good to know:** Browsing the SUN web site past weekend I found that SUN together with Barnes & Noble.com (one of the biggest online stores) offer to Java developers 10% discount on not discounted items! I thought it could be good to inform you about this offer:
  It is below:

  "July 2003
  Sun Developer Network and Barnes & Noble.com have joined forces to offer developers exclusive discounts on the latest Java books, bestsellers, DVDs and Music, all available on the Sun Developer Network online bookstore. Developers will save 10% off the list price of all non-discounted titles. This can add up to big savings for all the hot new Java books that you need. Free shipping is also offered (see site for details) and the discount is applied automatically at checkout. Start saving today!
  "
  Details are here: http://developers.sun.com/offers/

  P.S. I am not an affiliate to SUN (could be good) and it is not commercial announcement, just info for you :-)

- **Question:** What are the weak keys? And if they are weak why we need to such weak keys? Is not it better do not use weak things for such strong technology like Java? :-)

  Answer: This name has used for keys in WeakHashMap. An entry in a WeakHashMap will automatically be removed when its key is no longer in ordinary use. More precisely, the presence of a mapping for a given key will not prevent the key from being discarded by the garbage collector that is, made finalizable, finalized, and then reclaimed. When a key has been discarded its entry is effectively removed from the map, so this class behaves somewhat differently than other Map implementations.

  Both null values and the null key are supported. This class has performance

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

characteristics similar to those of the HashMap class, and has the same efficiency parameters of initial capacity and load factor.

- **Question:** I see that both Math and StrictMath contain methods for performing numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions... What is difference?

  Answer: You should use StrictMath to ensure portability of Java programs, the definitions of many of the numeric functions in this package require that they produce the same results as certain published algorithms.
  Unlike some of the numeric methods of class StrictMath, all implementations of the equivalent functions of class Math are not defined to return the bit-for-bit same results. This relaxation permits better-performing implementations where strict reproducibility is not required.

  By default many of the Math methods simply call the equivalent method in StrictMath for their implementation. Code generators are encouraged to use platform-specific native libraries or microprocessor instructions, where available, to provide higher-performance implementations of Math methods. Such higher-performance implementations still must conform to the specification for Math.

  The quality of implementation specifications concern two properties, accuracy of the returned result and monotonicity of the method. Accuracy of the floating-point Math methods is measured in terms of ulps, units in the last place. For a given floating-point format, an ulp of a specific real number value is the difference between the two floating-point values closest to that numerical value. When discussing the accuracy of a method as a whole rather than at a specific argument, the number of ulps cited is for the worst-case error at any argument. If a method always has an error less than 0.5 ulps, the method always returns the floating-point number nearest the exact result; such a method is correctly rounded. A correctly rounded method is generally the best a floating-point approximation can be; however, it is impractical for many floating-point methods to be correctly rounded. Instead, for the Math class, a larger error bound of 1 or 2 ulps is allowed for certain methods. Informally, with a 1 ulp error bound, when the exact result is a representable number the exact result should be returned; otherwise, either of the two floating-point numbers closest to the exact result may be returned.

  Besides accuracy at individual arguments, maintaining proper relations between the methods at different arguments is also important. Therefore, methods with more than 0.5 ulp errors are required to be semi-monotonic: whenever the mathematical function is non-decreasing, so is the floating-point approximation, likewise, whenever the mathematical function is non-increasing, so is the floating-point approximation. Not all approximations that have 1 ulp accuracy will automatically meet the monotonicity requirements.

  in this tip used API descriptions for Math and StrictMath classes

- **Question:** I work in Software Development many years and know that exist a lot of good and useful standards, standards "de facto" and simply unwritten rules for exchange of data between different OSs, programs and so on. For example XML... My question is: I know that different languages, platforms, OSs describe quite the same objects... I am wondering if exist any standard, agreement let say, how common objects are described... Part 1

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: The data, that you say describe all those common for whole SW industry objects, has own name - metadata. Metadata can be defined as information about data, or simply data about data. In practice, metadata is what most tools, databases, applications and other information processes use to define the structure and meaning of data objects.

One of the key issues limiting data interoperability today is that of incompatible metadata. The problem is that most applications define metadata using their own formats (schemas). This is compounded by the common practice of storing the metadata in different data management systems with different file formats. Without a common model for creating and accessing metadata, developers must hard-code discrete interfaces between applications to allow for data exchange and synchronization, thereby limiting interoperability and increasing the cost of developing and maintaining heterogeneous systems.

Standardizing on XML DTDs, which many industries are attempting to do, as a solution to this problem is insufficient, as DTDs do not have the capability to represent complex, semantically rich, hierarchical metadata.

Fortunately, Java has the solution for this problem - JMI (Java Metadata Interface) The JMI Specification implements a dynamic, platform-neutral infrastructure that enables the creation, storage, access, discovery, and exchange of metadata. JMI is based on the Meta Object Facility (MOF) specification from the Object Management Group (OMG), an industry-endorsed standard for metadata management. The MOF standard consists of a set of basic modelling artefacts described using UML. Models of any kind of metadata (called metamodels) can be built up from these basic building blocks. The MOF specification then defines a generic programming mechanism (using IDL) that allows for the discovery, query, access, and manipulation of these metamodel instances, either at design time or runtime. Thus the semantics of any modelled system can be completely discovered and manipulated.

JMI defines standard Java interfaces to these modelling components, and thus enables platform-independent discovery and access of metadata. These interfaces are generated automatically from the models for any kind of metadata. Additionally, metamodel and metadata interchange via XML is enabled by JMI technology's use of the XML Metadata Interchange (XMI) specification.

- **Question:** Part 2. What can YOU win by using JMI?

JMI will increase the adoption of standards-based metadata and accelerate the creation of applications and solutions in which there are no barriers to information exchange.
Applications using JMI can use the metadata to dynamically interpret the meaning of information, take action on that information and automate transactions across disparate systems and data sources.
used info from: http://java.sun.com/products/jmi/overview.html

- **Question:** (Java Bugs) Performance testing has shown that the finalize methods in the channel classes impose a significant GC penalty upon server-type programs that must deal with thousands of new open channels per second... (Java 1.4.1) What is solution?

Answer: The finalize methods of the primitive channel classes (SocketChannel,

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

ServerSocketChannel, etc.) have been removed (4777499). Performance testing revealed that the presence of these methods imposed a significant (factor of 100) garbage-collection penalty upon server-type applications that process thousands of new open channels per second. User code that deals in channels should be carefully written to close channels explicitly rather than rely upon the garbage collector to do so.

Tip based on doc "Enhancements and Changes Java 2 SDK, Standard Edition Version 1.4.2 "

- **Question:** What are changes for JNDI in Java 1.4.2?

  Answer: Bug 4639896 fixed: Java clients are now able to automatically follow referral using ldaps.

  The JNDI/LDAP service provider now supports LDAPS URLs for requesting access to an LDAP server using an SSL connection. LDAPS URLs are accepted anywhere LDAP URLs are accepted.

  In addition, if the user does not specify a port number in an LDAP URL, the default port is 389 for non-SSL connections and 636 for SSL connections. Prior to the Java 2 SDK, v 1.4.2, the port number defaults to 389 for all types of connections, including SSL.

  This change affects applications that use LDAP over SSL without explicitly specifying a port number and expect the default SSL port to be 389. 389 is the standard non-SSL port for LDAP so this configuration should be rare. The workaround is to explicitly specify the port number.

  Prior to the Java 2 SDK, v 1.4.2, the DNS service provider for JNDI did not include a trailing dot ('.') when returning a fully qualified domain name. Such a name could be returned within the value of an attribute, or by a call to Context.getNameInNamespace(). With the Java 2 SDK, v 1.4.2, the trailing dot is included.

- **Question:** Is it possible to get automatically updates of Java to my machine, like let say Norton Antivirus or Windows updates?

  Answer: Yes, if you install latest version of Java - 1.4.2_01 and later!
  Java Update is a new feature in the SDK/JRE. Its purpose is to provide the latest update of Java in a flexible, user-selectable manner. It works via a scheduler (jusched.exe) that runs as a background process and via the Java Plug-in Control Panel Update tab, which allows users to select the type of update options that they want.
  jusched.exe is launched when the user reboots the computer after installing the SDK/JRE. It runs as a background process, completely transparent to the user, but can be viewed in the Process tab of the Windows Task Manager. Should a user not want to have it running in the background, it can be killed via End Task.

  The Java Plug-in Control Panel Update tab allows the user to select the options for updating Java. The user can choose Automatic or Manual update. If the user chooses Automatic, then the update schedule (daily, weekly, monthly ...) can be selected via the Advanced button, and notification about updates can be set via the Notify Me:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

drop-down menu (before an update is downloaded and before it is installed, or only before it is installed). Manual update is performed by pressing the Update Now button. For more detail, see the Java Plug-in Control Panel Update tab.

- **Question:** The "java" command supports both options for pointing of classpath: -cp and -classpath. But the command "javac" recognizes only -classpath option. Why is such inconsistency?

  Answer: Not now, the Java 1.4.2 makes usage of "-classpath" synonym -cp possible.

- **Question:** I know that in Java exist deprecated Classes, methods, fields and so on. When I look at particular class description I can see that some method is deprecated. Can I get somewhere whole list of deprecated Java stuff?

  Answer: Yes! Every version of Java API, if you open for example framed version, contains in the right frame on the very beginning of page link "Deprecated". Just click it and you will be redirected to the page where all deprecated Classes, methods, fields, etc ... are listed.

- **Question:** What are maintenance releases of Java? And what are functionality releases?

  Answer: Maintenance releases (such as 1.4.1, 1.4.2) do not introduce any new language features or APIs, so they maintain source-compatibility with each other.

  Functionality releases and major releases (such as 1.3.0, 1.4.0) maintain upwards but not downwards source-compatibility.

- **Question:** Are all Java versions strongly compatible with previous versions of the Java 2 Platform? I have doubts, since sometime old code does not run on newer Java VMs... Part 1

  Answer: Yes, generally speaking Java always strongly compatible with previous versions of the Java 2 Platform. Almost all existing programs should run on latest Java version without modification. However, there are some minor potential incompatibilities that involve rare circumstances and "corner cases" that we are documenting here for completeness.

  If you have problems with running your old code on newer Java versions I recommend you check next pages where you can find all Java incompatibilities between Java versions:

  The following documents contain information about incompatibilities between adjacent releases.

  Incompatibilities in J2SE 1.4.1 (since 1.4.0) -
  http://java.sun.com/j2se/1.4.1/compatibility.html

  Incompatibilities in J2SE 1.4.0 (since 1.3) -
  http://java.sun.com/j2se/1.4/compatibility.html

  Incompatibilities in J2SE 1.3 (since 1.2) -
  http://java.sun.com/j2se/1.3/compatibility.html

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Incompatibilities in J2SE 1.2 (since 1.1) -
http://java.sun.com/products/jdk/1.2/compatibility.html

Question: Are all Java versions strongly compatible with previous versions of the Java 2 Platform? I have doubts, since sometime old code does not run on newer Java VMs... Part 2

Incompatibilities in JDK 1.1 (since 1.0) -
http://java.sun.com/j2se/index.jsp

Also, see the Java Language Specification Maintenance Page (http://java.sun.com/docs/books/jls/jls-maintenance.html) for a summary of changes that have been made to the specification of the Java programming language since the publication of the Java Language Specification, Second Edition (http://java.sun.com/docs/books/jls/index.html).

Just one example (Java 1.4.2):
Prior to J2SE 1.4.0, the character encoding scheme for security policy files was unspecified, and the files were read in using the default character encoding. Starting in 1.4.0, the policy files were required to be encoded in UTF-8.

While this allowed a policy file to be used across different locales, it broke existing policy files that contained characters in the default encoding.

In J2SE 1.4.2, a new system property was introduced: sun.security.policy.utf8

If this system property is set to true, the policy file is read in using UTF8 (1.4.0 and 1.4.1 behavior). If the system property is set to false, the policy file is read in using the default encoding (pre-1.4.0 behavior). When the system property is not set (it is NULL), its default value is true.

- **Question:** We are going to use Java on Linux. Are there any precautions should be taken against possible "well known" problems?

Answer: Unfortunately you should... Different releases of Java have different troubles...

For example:

A glibc 2.2 bug in versions 2.2.4 and earlier can cause the virtual machine to hang on exit when there is only one remaining active thread. This problem can affect Linux platforms such as Red Hat 7.0, 7.1, and 7.2 that have glibc versions 2.2 to 2.2.4. The problem is fixed in glibc 2.2.5.

I think that many programs can have "only one remaining active thread".
Every Java release usually lists "known" problems. For latest one, 1.4.2 it is here:

http://java.sun.com/j2se/1.4.2/install-linux.html#troubleshooting

It is recommendable to read the whole page since it contains very important info for those who first time starts with Java on Linux:

System Requirements

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Java 2 SDK Installation Instructions
Installation of Self-Extracting Binary
Installation of RPM File
Java Plug-in Browser Registration Instructions
Java Web Start Installation Notes
Troubleshooting
Known Problems on Non-Supported Platforms

- **Question:** In Java API I found ACL and AclEntry interfaces, but do not understand what ACL is exactly used for...

  Answer: ACL come from Solaris and was used to increase security level for file access. Beside user and groups in one of latest versions of Solaris (I think Solaris 7 or 8) was added ACL feature.

  ACLs enable you to define file or folder permissions for the owner, owner's group, others, and specific users and groups, and default permissions for each of these categories. You can set up only one ACL per file or folder. An ACL consists of ACL entries. Each entry has a user type associated with it, much as basic permissions have Owner, Group, or Other associated with them.

  If you want to grant file or folder access to a particular user or group, use the User or Group type ACL entry, respectively. You must specify the name of the user or group when you create the ACL entry. For example, you can use an ACL to grant a friend read permission on your resume, while protecting it from being read by the rest of the world (besides yourself).

  ACL-enabled files and folders have a mask defined whose default permissions are the group permissions for the file or folder. The mask is the maximum allowable permissions granted to any user on all ACL entries and for Group basic permissions. It does not restrict Owner or Other basic permissions. For example, if a file's mask is read-only, then you cannot create an ACL with write or execute permission for a user without changing the mask value.

  Use the mask as a quick way to limit permissions for users and groups.
  More information can be found on http://docs.sun.com

  I do not know if other UNIX's also uses ACL concept. Could be good if someone informed me about it :-)

- **Question:** Often when I see the name of some Java technology, it followed by brackets with "JSR" and some number. For example: "J2ME Web Services version 1.0, (JSR-172) "
  What does it mean - JSR-172?

  Answer: JSRs are Java Specification Requests (JSRs). JSRs are the actual descriptions of proposed and final specifications for the Java platform. At any one time there are numerous JSRs moving through the review and approval process in JCP (Java Community Process).

  You can see a list of all JSRs, by visiting the Java Specification Requests list (http://jcp.org/en/jsr/all). Using the links near the top of the page, you can sort the list by JSR ID number, Title, and the Spec Lead Company Name.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The JCP (Java Community Process) holds the responsibility for the development of Java technology. As an open, inclusive organization of active members and non-member public input, it primarily guides the development and approval of Java technical specifications. Anyone can join the JCP and have a part in its process, and you don't even have to join to contribute as a public participant.

The work of the Java Community under the JCP's procedures helps to ensure Java technology's standard of stability and cross-platform compatibility, enabling it to operate on hundreds of millions of devices, from desktop computers to consumer electronics to industrial robots.

Just as important, the JCP continually grows the platform's specification portfolio to meet the emerging technology needs of developers and organizations globally that depend on Java technology.

Like to get involved? Anyone can join the JCP and have a part in its process, and you don't even have to join to contribute as a public participant. Here are the pages that will help you become a member of the JCP:
http://jcp.org/en/introduction/overview

You can ask me: Why do I need this formal stuff? You can get most fresh information about coming Java technologies and plan your future developments. If you know that some technology, that you need, comes next month then you can concentrate on another things. And use it when it comes, instead of trying to develop your own solution...

Ł **Question:** Java 1.4.1 introduced two new garbage collectors. You described shortly them, in one of your tips. Could you give more info on when and which GC is preferable to use? Part 1

Answer: In the J2SE platform version 1.4.1 two new garbage collectors were introduced to make a total of four garbage collectors from which to choose. In J2SE platform version 1.4.2 the performance of the new collectors have been improved through algorithm optimizations and bug fixes, along with documentation updates to educate developers and administrators on what collectors to choose and when.

Throughput Collector
The throughput collector uses a parallel version of the young generation collector. It is used by passing the -XX:+UseParallelGC on the command line. The tenured generation collector is the same as the default collector. Use the throughput collector when you want to improve the performance of your application with larger numbers of processors.

In the default collector garbage collection is done by one thread, and therefore garbage collection adds to the serial execution time of the application. The throughput collector uses multiple threads to execute a minor collection and so reduces the serial execution time of the application.

Concurrent Low Pause Collector
The concurrent collector is used to collect the tenured generation and does most of the collection concurrently with the execution of the application. The concurrent collector employs a separate collector thread that consumes CPU cycles during application execution, this allows the application to be paused for only short periods of time during the collection but could lower overall throughput. It is used by passing

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

the -XX:+UseConcMarkSweepGC on the command line.

Use the concurrent collector if your application would benefit from shorter garbage collector pauses and can afford to share processor resources with the garbage collector when the application is running. Typically applications which have a relatively large set of long-lived data (a large tenured generation), and run on machines with two or more processors tend to benefit from the use of this collector. However, this collector should be considered for any application with a low pause time requirement. Optimal results have been observed for interactive applications with tenured generations of a modest size on a single processor.

- **Question:** Java 1.4.1 introduced two new garbage collectors. You described shortly them, in one of your tips. Could you give more info on when and which GC is preferable to use? Part 2

  Also exist "AggressiveHeap" option to be used on Servers, to improve performance. It was originally intended for machines with large amounts of memory and a large number of CPUs, but in the J2SE platform, version 1.4.1 and later it has shown itself to be useful even on four processor machines. The physical memory on the machines must be at least 256MB before AggresiveHeap can be used.
  Detailed recommendations (with tests, graphs, etc.) can be found here:
  http://java.sun.com/docs/hotspot/gc1.4.2/index.html

- **Question:** It is very difficult to tune Garbage Collection for large systems (much memory, many CPUs) properly from first run and without experience. It can take much time to find most optimized configurations. Does exist any kind intelligent GCing which is based on some history of application run, so say self tuning?

  Answer: Yes, if you use adaptive sizing, which is actually enabled by default. We just do not know it :-)
  A feature available with the throughput collector in the J2SE platform, version 1.4.1 and later releases is the use of adaptive sizing (-XX:+UseAdaptiveSizePolicy), which is on by default. Adaptive sizing keeps statistics about garbage collection times, allocation rates, and the free space in the heap after a collection. These statistics are used to make decisions regarding changes to the sizes of the young generation and tenured generation so as to best fit the behavior of the application. Use the command line option -verbose:gc to see the resulting sizes of the heap.

- **Question:** Does name "Concurrent Low Pause Collector" (garbage collector) mean that this collector runs very fast in very short time when application paused shortly (low paused)?

  Answer: Quite the opposite! "Low Pause" means that GC runs almost always and GC has short pauses. This kind of GC is supposed to share computer resources with the application and run concurrently. Optimal results have been observed for interactive applications.

- **Question:** What is floating garbage and what should I do with it? Or not?

  Answer: Since garbage collector and your application run concurrently on the machine, some part of objects that were alive in the beginning of GCing may become dead when GCing finishes. You should not worry about it. Although the amount of floating garbage is ~20% from all garbage (quite much) it will be collected by

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

garbage collector at the next pass.

- **Question:** Why do we need wrappers like Integer, Boolean for int, boolean in Java?

  Answer: Because wrappers let us treat primitives as objects. For example, you can not add an int to Vector, but you can the instance of Integer.

  P.S. It is very short answer. If you know more please send us your explanation and we will publish it!

- **Question:** I read that JDK 1.x has new features...
  I am not quite sure about the stability and performance. Does it mean that 1.x is not suitable for development now?

  Answer: Absolutely not. When you are first starting out on a new project, especially big one it is a good idea to work with the most recent version of something. By the time you are ready to ship, the version of software that you use will have had the bugs shaken out of it. And you can include it into advantages of your product saying that it is built on latest software platform :-)

- **Question:** Is there any way of getting the memory requirements of objects at run-time?

  Answer: JDK 1.1 provides a -prof option which among other things prints the bytes per instance and total instances for every class loaded in the VM at termination.

  Java 2 provides a heap profiler which gives a detailed map of heap usage at termination.
  http://developer.java.sun.com/developer/onlineTraining/Programming/JDCBook/perf3.html

- **Question:** I'm just getting into Java and have a simple **Question:**
  What is J2EE and how is it different from Java2?'

  Answer: Java2 is the name of the current specification of the Java language, the JVM
  and all the other things that make up the basics of Java.

  Java2 is a specification. It is implemented in the product SDK 1.3 now and soon will be available in 1.4 version (beta now).

  Java has 3 subsections:

  1. J2EE - Enterprise Edition - for server applications and other back-end...

  2. J2SE - Standard edition - for desktop use...

  3. J2ME - Micro edition - runs in cell phones, PDAs, and other resource and

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

memory constrained devices, on the order of a 16 bit RISC processor and 128-512k RAM.

The 3 sections listed above are not products - they are specs, and define the tools available, language semantics, standard APIs and optional APIs.

J2ME is a subset of the Java2 spec.
J2EE is a superset of J2SE (+ extra APIs, + functionality).
The difference between J2EE and J2SE is J2EE contains extra APIs
A J2EE application is built using the J2SE tools, the
difference is the extra APIs.

Ł **Question:** When do I need to use reflection?

Answer: Reflection is a feature in the Java programming language. It allows an executing Java program to examine or "introspect" upon itself, and manipulate internal properties of the program. For example, it's possible for a Java class to obtain the names of all its members and display them.

The ability to examine and manipulate a Java class from within itself may not sound like very much, but in other programming languages this feature simply doesn't exist. For example, there is no way in a Pascal, C, or C++ program to obtain information about the functions defined within that program.

The reflection API represents, or reflects, the classes, interfaces, and objects in the current Java Virtual Machine. You'll want to use the reflection API if you are writing development tools such as debuggers, class browsers, and GUI builders. With the reflection API you can:

Determine the class of an object.

Get information about a class's modifiers, fields, methods, constructors, and superclasses.

Find out what constants and method declarations belong to an interface.

Create an instance of a class whose name is not known until runtime.

Get and set the value of an object's field, even if the field name is unknown to your program until runtime.

Invoke a method on an object, even if the method is not known until runtime.

Create a new array, whose size and component types is not known until runtime, and then modify the array's components.

Read more about reflection here:
http://developer.java.sun.com/developer/technicalArticles/ALT/Reflection/
http://java.sun.com/docs/books/tutorial/reflect/index.html

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** Properties class inherited from Hashtable...
Does it mean that I can use "non-string" keys and values?

Answer: Not really. Why?
I think best answer for this question can be found in API:

"The Properties class represents a persistent set of properties. The Properties can be saved to a stream or loaded from a stream. Each key and its corresponding value in the property list is a string.

A property list can contain another property list as its "defaults"; this second property list is searched if the property key is not found in the original property list. Because Properties inherits from Hashtable, the put and putAll methods can be applied to a Properties object. Their use is strongly discouraged as they allow the caller to insert entries whose keys or values are not Strings. The setProperty method should be used instead. If the store or save method is called on a "compromised" Properties object that contains a non-String key or value, the call will fail.

When saving properties to a stream or loading them from a stream, the ISO 8859-1 character encoding is used. For characters that cannot be directly represented in this encoding, Unicode escapes are used; however, only a single 'u' character is allowed in an escape sequence. The native2ascii tool can be used to convert property files to and from other character encodings."

Question: Why do we need to use "Preferences" if we already have "Properties" and JNDI in Java?

Answer: Prior to the introduction of the Preferences API, developers could choose to manage preference and configuration data in an ad hoc fashion, by using the Properties API or the JNDI API as described below.

Often, preference and configuration data was stored in properties files, accessed through the java.util.Properties API. However, there are no standards as to where such files should reside on disk, or what they should be called. Using this mechanism, it is extremely difficult to backup a user's preference data, or transfer it from one machine to another. As the number of applications increases, the possibility of file name conflicts increases. Also, this mechanism is of no help on platforms that lack a local disk, or where it is desirable that the data be stored in an external data store (such as an enterprise-wide LDAP directory service).

Less frequently, developers stored user preference and configuration data in a directory service, accessed through the Java Naming and Directory Interface (JNDI) API. Unlike the Properties API, JNDI allows the use of arbitrary data stores (back-end neutrality). While JNDI is extremely powerful, it is also rather large, consisting of 5 packages and 83 classes. JNDI provides no policy as to where in the directory name space the preference data should be stored, or in which name space.

Neither Properties nor JNDI provide a simple, ubiquitous, back-end neutral preferences management facility. The Preferences API does provide such a facility, combining the simplicity of Properties with the back-end neutrality of JNDI. It provides sufficient built-

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

in policy to prevent name clashes, foster consistency, and encourage robustness in the face of inaccessibility of the backing data store.

From SUN's Preferences API description

Question: How does this Preferences API relate to Properties API?

Answer: It is intended to replace most common uses of Properties, rectifying many of its deficiencies, while retaining its light weight. When using Properties, the programmer must explicitly specify a pathname for each properties file, but there is no standard location or naming convention. Properties files are "brittle", as they are hand-editable but easily corrupted by careless editing. Support for non-string data types in properties is non-existent. Properties cannot easily be used with a persistence mechanism other than the file system. In sum, the Properties facility does not scale.

From Preferences Design FAQ

Question: How does Preferences API relate to JNDI?

Answer: Like JNDI, it provides back-end neutral access to persistent key-value data. JNDI, however, is far more powerful, and correspondingly heavyweight. JNDI is appropriate for enterprise applications that need its power. Preferences API is intended as a simple, ubiquitous, back-end neutral preferences-management facility, enabling any Java application to easily tailor its behavior to user preferences and maintain small amounts of state from run to run.

From Preferences Design FAQ

Question: Preferences: Why do all of the get methods require the caller to pass in a default?

Answer: This forces the application authors to provide reasonable default values, so that applications have a reasonable chance of running even if the repository is unavailable.

From Preferences Design FAQ

Question: Why doesn't this API provide stronger guarantees concerning concurrent access by multiple VMs? Similarly, why doesn't the API allow multiple Preferences updates to be combined into a single "transaction", with all or nothing semantics?

Answer: Preferences: While the the API does provide rudimentary persistent data storage, it is not intended as a substitute for a database. It is critical that it be possible to implement this API atop standard preference/configuration repositories, most of which do not provide database-like guarantees and functionality. Such repositories have proven adequate for the purposes for which this API is intended.

From Preferences Design FAQ

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Question: Preferences: Why doesn't this API contain methods to read and write arbitrary serializable objects?

Answer: Serialized objects are somewhat fragile: if the version of the program that reads such a property differs from the version that wrote it, the object may not deserialize properly (or at all). It is not impossible to store serialized objects using this API, but we do not encourage it, and have not provided a convenience method.

From Preferences Design FAQ

Question: Why is Preferences an abstract class rather than an interface?

Answer: It was decided that the ability to add new methods in an upward compatible fashion outweighed the disadvantage that Preferences cannot be used as a "mixin" (That is to say, arbitrary classes cannot also be made to serve as Preferences objects.) Also, this obviates the need for a separate class for the static methods. (Interfaces cannot contain static methods.)

From Preferences Design FAQ

Question: Can I create different preferences for different users and at the same time have some "common", OS based preferences that are the same for all?

Answer: There are two separate trees of preference nodes, one for user preferences and one for system preferences. Each user has a separate user preference tree, and all users in a given system share the same system preference tree. The precise description of "user" and "system" will vary from implementation to implementation. Typical information stored in the user preference tree might include font choice, colour choice, or preferred window location and size for a particular application. Typical information stored in the system preference tree might include installation configuration data for an application.

From Preferences API

Question: Are methods that modify preferences asynchronous or synchronous?

Answer: All of the methods that modify preferences data are permitted to operate asynchronously; they may return immediately, and changes will eventually propagate to the persistent backing store with an implementation-dependent delay. The flush method may be used to synchronously force updates to the backing store. Normal termination of the Java Virtual Machine will not result in the loss of pending updates -- an explicit flush invocation is not required upon termination to ensure that pending updates are made persistent.

From Preferences API

- **Question:** What does mean term "peer" and what can I do with it if I use getPeer() in Component?

Answer: You should not use it now since it is deprecated.
Thee peer is a system dependant component. All "heavyweight" graphical components have a respective peer that is responsible for communication between

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

OS and Java. The peer level is different for different platforms but it makes possible to have "write once run anywhere" principle.
Swing almost does not use heavyweight components and you should not use it as well until you are going to adapt Java to new OS or write your own component...

- **Question:** I have developed an application. At present the application is in English Language.
Now I need to convert this site to another languages but don't know how to do it. Can anybody tell me the way to do it... I do not want to rewrite every time my code...

Part 1.

Answer: The best way is to use Resource Bundles. Keep different language versions in resource bundle file. Your program can load it from the resource bundle that is appropriate for the current user's locale.

This allows you to write programs that can:
be easily localized, or translated, into different languages handle multiple locales at once be easily modified later to support even more locales.

For example, the base name of a family of resource bundles might be "MyResources". The family should have a default resource bundle which simply has the same name as its family - "MyResources" - and will be used as the bundle of last resort if a specific locale is not supported. The family can then provide as many locale-specific members as needed, for example a German one named "MyResources_de".

Each resource bundle in a family contains the same items, but the items have been translated for the locale represented by that resource bundle. For example, both "MyResources" and "MyResources_de" may have a String that's used on a button for canceling operations. In "MyResources" the String may contain "Cancel" and in "MyResources_de" it may contain "Abbrechen".

If there are different resources for different countries, you can make specializations: for example, "MyResources_de_CH" contains objects for the German language (de) in Switzerland (CH). If you want to only modify some of the resources in the specialization, you can do so.

Question: I have developed an application. At present the application is in English Language.

Part 2.
When your program needs a locale-specific object, it loads the ResourceBundle class using the getBundle method:

```
ResourceBundle myResources =
                    ResourceBundle.getBundle("MyResources", currentLocale);
```

Resource bundles contain key/value pairs. The keys uniquely identify a locale-specific object in the bundle. Here's an example of a ListResourceBundle that contains two key/value pairs:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public class MyResources extends ListResourceBundle {
   public Object[][] getContents() {
      return contents;
  }
  static final Object[][] contents = {
      // LOCALIZE THIS
      {"OkKey", "OK"},
      {"CancelKey", "Cancel"},
      // END OF MATERIAL TO LOCALIZE
   };
}
```

Keys are always Strings. In this example, the keys are "OkKey" and "CancelKey". In the above example, the values are also Strings--"OK" and "Cancel"--but they don't have to be. The values can be any type of object.

You retrieve an object from resource bundle using the appropriate getter method. Because "OkKey" and "CancelKey" are both strings, you would use getString to retrieve them:

```
button1 = new Button(myResources.getString("OkKey"));
button2 = new Button(myResources.getString("CancelKey"));
```

The getter methods all require the key as an argument and return the object if found. If the object is not found, the getter method throws a MissingResourceException.
Besides getString, ResourceBundle also provides a method for getting string arrays, getStringArray, as well as a generic getObject method for any other type of object.

Question: I have developed an application. At present the application is in English Language.

Part 3

When using getObject, you'll have to cast the result to the appropriate type. For example:

```
int[] myIntegers = (int[]) myResources.getObject("intList");
```

The Java 2 platform provides two subclasses of ResourceBundle, ListResourceBundle and PropertyResourceBundle, that provide a fairly simple way to create resources. As you saw briefly in a previous example, ListResourceBundle manages its resource as a List of key/value pairs. PropertyResourceBundle uses a properties file to manage its resources.

If ListResourceBundle or PropertyResourceBundle do not suit your needs, you can write your own ResourceBundle subclass. Your subclasses must override two methods: handleGetObject and getKeys().

- **Question:** I have developed an application. At present the application is in English Language.
Now I need to convert this site to another languages but don't know how to do it. Can anybody tell me the way to do it... I do not want to rewrite every time my code...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Part 4

Answer: The following is a very simple example of a ResourceBundle subclass, MyResources that manages two resources (for a larger number of resources you would probably use a Hashtable). Notice that you don't need to supply a value if a "parent-level" ResourceBundle handles the same key with the same value (as for the okKey below).

Example:

```
// default (English language, United States)
public class MyResources extends ResourceBundle {
  public Object handleGetObject(String key) {
    if (key.equals("okKey")) return "Ok";
    if (key.equals("cancelKey")) return "Cancel";
    return null;
  }
}

// German language
public class MyResources_de extends MyResources {
  public Object handleGetObject(String key) {
    // don't need okKey, since parent level handles it.
    if (key.equals("cancelKey")) return "Abbrechen";
    return null;
  }
}
```

You do not have to restrict yourself to using a single family of ResourceBundles. For example, you could have a set of bundles for exception messages, ExceptionResources (ExceptionResources_fr, ExceptionResources_de, ...), and one for widgets, WidgetResource (WidgetResources_fr, WidgetResources_de, ...); breaking up the resources however you like.

This tip is based on API description.

Question: Can anyone tell me why the Java Vector class is called a "Vector"? I come from the engineering world, so I am trying to make some sense out of the name.

Answer: "Vector" is a common name for an array of one dimension. I think that it was designed to be accessed in only one direction - from start to end.
Of course, you can write code to iterate backwards...
But the enumeration it goes you don't have a getPrev() method, only a getNext():

```
for (Enumeration e = v.elements() ; e.hasMoreElements() ;) {
  System.out.println(e.nextElement());
}
```

So a Vector has dimension (number of elements) and direction (in terms of access). Are you agreeing?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** How is a multidimensional array stored in Java? Is it different from C?

  Answer: Java stores multidimensional arrays as single dimension arrays which consist themselves of more single dimensional arrays.

- **Question:** Why do we need to use shift operator? Can you give some examples?

  Answer:

  1. Make faster integer division/multiplication operations:
  4839534 * 4
  can be done like this:
  4839534 << 2
  or
  543894 / 2
  can be done like this:
  543894 >> 1
  Shift operations much more faster than multiplication for most of processors.

  2. Reassembling byte streams to int values
  3. For accelerating operations with graphics since Red, Green and Blue colors coded by separate bytes.
  4. Packing small numbers into one single long...

- **Question:** For example I use some collection and method put() which has return type Object... I do not need it and do not use it in my program.
  Does this Object occupy memory space?

  Answer: If it is not referenced - it will be garbage collected. And it will stay in memory in opposite case.
  The time of garbage collection is not known in advance - it is up to GC (Garbage Collector) to decide when it can be done.

- **Question:** Why I can not compile it:

  ```
  byte aa = 2;
  aa = - - aa;
  ```

  I think the construction: - -aa is equivalent to -(-aa)...

  Answer: The Java programming language supports various arithmetic operators for all floating-point and integer numbers. These operators are + (addition), - (subtraction), * (multiplication), / (division), and % (modulo).

  In your example aa is a byte variable. You can use --aa instead.

  Interesting fact:

  probably you know that the Class name you can get like this:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
MyClass m = new MyClass();
System.out.println(m.getClass());
```

Another not documented way is:

```
MyClass m = new MyClass();
System.out.println(MyClass.class);
```

The interesting fact is that you will not find "class" field in Class or Object!

- **Question:** What does mean "internalization" for me when I design my program and where should I go if I need to change the language in some program?

  Answer: I think the best Answer can be found on SUN's site :-):

  "Internationalization is the process of designing an application so that it can be adapted to various languages and regions without engineering changes. Sometimes the term internationalization is abbreviated as i18n, because there are 18 letters between the first "i" and the last "n."

  An internationalized program has the following characteristics:

  - With the addition of localized data, the same executable can run worldwide.
  - Textual elements, such as status messages and the GUI component labels are not hard coded in the program. Instead they are stored outside the source code and retrieved dynamically.
  - Support for new languages does not require recompilation.
  - Culturally-dependent data, such as dates and currencies, appear in formats that conform to the end user's region and language.
  - It can be localized quickly.
  - Localization is the process of adapting software for a specific region or language by adding locale-specific components and translating text. The term localization is often abbreviated as l10n, because there are 10 letters between the "l" and the "n."

  Usually, the most time-consuming portion of the localization phase is the translation of text. Other types of data, such as sounds and images, may require localization if they are culturally sensitive. Localizers also verify that the formatting of dates, numbers, and currencies conforms to local requirements."

  Please check a quick example and checklist (what you need to do...):

  http://java.sun.com/docs/books/tutorial/i18n/intro/index.html

- **Question:** My program creates a huge number of objects dynamically and even more inside each of them...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

One thread does some processing and when the user calls a method on the object, it will kill the thread.

After that the object has to be killed so that it can be garbage collected...

How do I do it?

Answer: You do not "kill" objects, just stop just stop keeping any references to them and garbage collector will do the rest... :-)

- **Question:** Can I force finalization using System.runFinalization()?

Answer: No, in Java API is written:

"Calling this method suggests that the Java Virtual Machine expend effort toward running the finalize methods of objects that have been found to be discarded but whose finalize methods have not yet been run. When control returns from the method call, the Java Virtual Machine has made a best effort to complete all outstanding finalizations. "

It means that you just suggest the JVM that it might be a good idea to do finalization...

Like garbage collection, you cannot force it.

- **Question:** I read that an object can be strongly, weakly or even phantom (!!!!) reachable...???

Answer: I found the definitions in API. I think it is clear enough to be cited here:

Going from strongest to weakest, the different levels of reachability reflect the life cycle of an object. They are operationally defined as follows:

An object is strongly reachable if it can be reached by some thread without traversing any reference objects. A newly-created object is strongly reachable by the thread that created it.

An object is softly reachable if it is not strongly reachable but can be reached by traversing a soft reference.

An object is weakly reachable if it is neither strongly nor softly reachable but can be reached by traversing a weak reference.

When the weak references to a weakly-reachable object are cleared, the object becomes eligible for finalization.

An object is phantom reachable if it is neither strongly, softly, nor weakly reachable, it has been finalized, and some phantom reference refers to it.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Finally, an object is unreachable, and therefore eligible for reclamation, when it is not reachable in any of the above ways.

About phantom reference please read more in API (java.lang.ref.PhantomReference)

- **Question:** I use logging in my application... The problem is that the problem is that I am opening the PrintWriter stream only once and keep it open until my application is stopped. Open PrintWriter stream does not let to open logs in text editor and users can not read logs... Appears a error message like this: "Can not open file, it used by another application..."

  Is it possible to solve the problem?

  Answer: It depends on OS type. For example in Windows NT you can use command "type":

  type log.txt > log_copy.txt and then read log_copy.txt file in any text editor...

  In UNIX use command "cat":

  cat log.txt > log_copy.txt

- **Question:** I know that exist many Java compilers, not only "javac" from SUN :-)

  Part 1
  Could you list some of them and describe advantages of them over SUN's version?

  Answer: I list some of them here with short descriptions... Most of them are extending Java with additional features, keywords, functions and so on..

  1. Java espresso is a Java 1.0 compliant compiler being developed at Boston University:
  http://types.bu.edu/Espresso/JavaEspresso.html

  The objectives of this implementation were to understand the issues involved in the development of a modern object-oriented compiler for a real language, to study the Java Virtual Machine (JVM) and its runtime system

  2. The GNU Compiler for the Java: http://gcc.gnu.org/java/
  GCJ is a portable, optimizing, ahead-of-time compiler for the Java Programming Language. It can compile:
  Java source code directly to native machine code,
  Java source code to Java bytecode (class files),
  and Java bytecode to native machine code.

  3. The Jikes compiler:
  http://oss.software.ibm.com/developerworks/opensource/jikes/
  Jikes has four advantages that make it a valuable contribution to the Java community:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- Open source. Jikes is OSI Certified Open Source Software. OSI Certified is a certification mark of the Open Source Initiative.

- Strictly Java compatible. Jikes strives to adhere to both the Java Language Specification and the Java Virtual Machine Specification as tightly as possible, and does not support subsets, supersets, or other variations of the language. The FAQ describes some of the side effects of this strict language conformance.

- High performance. Jikes is a high performance compiler, making it ideal for use with larger projects.

- Dependency analysis. Jikes performs a dependency analysis on your code that provides two very useful features: Incremental builds and makefile generation.

--> to be continued in our next newsletter -->

Ł **Question:** I know that exist many Java compilers, not only "javac" from SUN :-)

Part 2.

Could you list some of them and describe advantages of them over SUN's version?

4. The JUMP compiler:

http://ourworld.compuserve.com/homepages/DeHoeffner/jump.htm

JUMP extends the capabilities of JAVAC in the following ways:

It is about 10 - 30 times faster than JAVAC.
The output format can be changed to adapt JUMP to your favourite development environment.
It contains a fully functional disassembler (option -p) that can be used as a replacement for the JAVAP disassembler that ships with Suns JDK.
It extends the JAVA language with several powerful features well known from C++. These features include class templates, operator overloading and default parameters. They are well accepted in the C++ community and have been proofed to leverage productivity of the developers.

5. The KOPI compiler: http://www.dms.at/kopi/index.html

The Kopi Project is a Java software project from DMS, providing a development environment for database applications using Java, JDBC and JFC/Swing.

6. The Pizza Compiler: http://pizzacompiler.sourceforge.net/

The Pizza language is an extension to Java with three new features:
- Generics (a.k.a. Parametric polymorphism)
- Function pointers (a.k.a. First-class functions)
- Class cases and pattern matching (a.k.a. Algebraic types)

7. The Timber compiler: http://www.pds.twi.tudelft.nl/timber/index.html

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The Timber compiler has been developed to demonstrate, and experiment with, compilation techniques for parallel and distributed systems.

The compiler implements most of Java, plus a set of additional language constructs for scientific programming. This set of extensions is called Spar. We only fundamentally deviate from the Java language specification on the following points:

The compiler is a whole-program static compiler from Java/Spar source code to machine code, so JVM bytecode is not supported. In particular, dynamic class loading is not supported.

Java threads are not supported. Instead, we provide our own language constructs for parallel programming.

The Spar language extensions introduce a number of additional keywords.
--
John

- **Question:** I am writing an application for small memory system. To make it reliable I would like to monitor memory size in my program and try to decrease program activity to enable Garbage Collector do its job before all memory will be consumed.

  When enough memory will be available program must continue the job.

  How can I do that?

  Answer: You idea is excellent to design long time running program for an extreme environment :-)

  Use Runtime.totalMemory() and Runtime.freeMemory()

- **Question:** How can I store and retrieve serialized objects to and from a file?
  I have a .txt file and two String objects say,

  String company = "Company",

  String company_name = "Comp_name",

  How do I that?

  Answer: I found one example how to write a serialized object to a file, tmp. The file tmp will be in the same directory as the program:

```
import java.io.*;
import java.util.*;

// Writing an object.
// Serialize today's date to a file.

public class WriteData implements Serializable {
  public static void main (String[] args) throws IOException {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    FileOutputStream f = new FileOutputStream("tmp");
    ObjectOutputStream s = new ObjectOutputStream(f);
    String company = "My Good Company";
    s.writeObject(company);
    s.flush();
  }
}
```

Here is an example of how to read a serialized object from the file "tmp":

```
import java.io.*;
import java.util.*;
import java.lang.*;

// Reading an Object from a stream.
// Deserialize a String and date from a file.

public class ReadData implements Serializable {
  public static void main (String[] args)
      throws IOException,ClassNotFoundException {
    FileInputStream in = new FileInputStream("tmp");
    ObjectInputStream s = new ObjectInputStream(in);
    String company = "";
    String company_name = "";
    company = (String)s.readObject();
    System.out.println(company + " ");
  }
}
```

- **Question:** Do you know how can I display special symbols like "copyright" & "trademark"???
  What are theirs codes?

  Answer: Please take a look at:
  Code Charts - http://www.unicode.org/charts/
  or
  Unicode Character Name Index
  http://www.unicode.org/charts/charindex.html

- **Question:** Why is it better to write own close() method than rely to finalize()?

  Answer: Finalize() called by the garbage collector on an object when garbage collection determines that there are no more references to the object. A subclass overrides the finalize method to dispose of system resources or to perform other cleanup.

  There are two things bad with finalize:
  1. Garbage collection might not ever run...
  2. Finalize order is not predictable when you have multiple objects

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

The more correct approach is to write close() methods for each object.
Also if some exception can be thrown inside of method that calls close() method use finally. It fits best for solving such problems.
No matter how try block exits - either normally or exceptionally - you will be able to run finally statement.

- **Question:** If a class inherited from an interface, can super() be used in method of the class? An interface has no concrete method... Very strange...
How can super() be used?'

  Answer: Super() method appeals to the constructor of parent class that implements interface, not to "parent" interface itself.
If there was no parent class then java.lang.Object will be automatically extended and the default constructor from the Object class will be called.

- **Question:** Why do we need wrappers like Integer, Boolean for int, boolean in Java?

  Answer: Because wrappers let us treat primitives as objects. For example, you can not add an int to Vector, but you can the instance of Integer.

  P.S. It is very short answer. If you know more please send us your explanation and we will publish it!

- **Question:** If there was a "setvalue()" method in Float class, it would have been useful to easily create code that modify parameters passed on methods.
So any good reason why there are no setvalue methods?

  Answer: Once a Float instance is created you can not change its value because it is immutable by definition. The rest of wrappers are also immutable, for example Integer, Boolean and so on.
Why? It has two main advantages...
Nobody can change the instance and cause bugs in program.
Also they are thread safe: if nobody can change them why we need to protect them?
If you need to have mutable wrappers you can probably write them and use for your own needs. But you can not do anything with classes from third party companies that use standard implementation.

- **Question:** When I running Runtime.exec(...), I get a Process object.
I want serialize it to a file to be able to reconstruct it later if required... Is it possible?

  Answer: no, it is not possible.

- **Question:** My application uses just one class from big jar file. Does the entire jar file get loaded into memory?

  Answer: No. Classloader indexing once whole content of jar file and loads the list of content into memory. The necessary class will be loaded when it is called, in so

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

called "lazy" fashion...

- **Question:** Why is thin client has name "thin" in J2EE?

  Answer: J2EE applications do not perform most of functions such as executing complex logic, querying databases on a client side. Instead, these operations are executed on J2EE server by enterprise or Web beans. It has advantages in terms of security, performance, reliability and services. Main function of a thin client is to provide some interface to users, letting to interact with J2EE application on a server side.

- **Question:** What is difference between the for(;;) and while(true)?

  Answer: There is no difference! They both are infinite loops... Loop forever - Perpetum Mobile (Latin) :-)

- **Question:** I just started to work in big company... Before I worked with software design myself and developed small programs. Now I hear such terms like "business layer", "presentation layer"... Has it something to do with Java? Please clarify.

  Answer: It has more to do with applications structure and used for design of large software products. It helps to divide application logic into big logically independent blocks (layers) that interacts by interfaces. It lets to develop big software products without knowledge of all details.

  We will describe here three layers model:

  1. Presentation layer - responsible for content displaying to user and provides interface for user - application interaction. It is client in client - server application.

  2. Business layer - main logic of an application is implemented here: all required calculations, data validations and so on. Usually runs on server side in client - server application.

  3. Data layer - responsible for data management. This layer provides business layer with all data that are processed there and then presented to user by presentation layer.

  Layering of an application lets to change different parts without significant efforts. For example if data layer operates with a SQL database it can be switched to Oracle database. Well planned and structured application will let do it easy.

- **Question:** Java has no sizeof() like C++... Why?

  Answer: In C and C++ sizeof() tells the number of allocated bytes for data items. Different machines and OSs can allocate different number of bytes for the same data. For example integer can be stored in 16 or 32 bits. That's why important to know for programmer this information.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Java is much better in this situation since it keeps the same size on all machines. The portability is a part of language and you should not worry about this - just know it!
--
Sasha

- **Question:** I'm trying to write a script to package my class files into a jar file. I can use "jar cf a.jar temp" and it takes all files in the temp directory. Unfortunately it also includes all source java files. Is it possible to just include the *.class files but also recourse all directories?

  Answer: Compile with the -d option that lets to specify a directory where your class files must be placed.

- **Question:** I've been using the java comm API to allow me to build a small java interface to an old datalogger box that connects to a computer's serial port. The program runs and returns data to the interface. I have no problem receiving data from the datalogger.

  However, the program that runs on the datalogger needs to be reset with a "cbreak," that is, the equivalent of sending it ^C (control-C).
  If I use a terminal program to send the cbreak (by typing ctrl-c), it works fine, resetting the datalogger. What I can't figure out is how to get java to send the cbreak, specifically, what do I need to write that the SerialPort's Outstream? Is it a special character? I've tried sending the string "^C" but that doesn't work.

  I assume it must be some Unicode thing that I am not competent enough to deal with.

  Answer: No, in fact control-c has the numerical value 3. Just send a byte with a numerical value of 3.
  --
  Paul Lutus
  www.arachnoid.com

- **Question:** I want to download Java 1.4. I looked on the page for compatibility and did not find anything about Windows XP.

  Has anyone installed 1.4 on XP? I want to be sure before doing this...

  Answer: Yes, it seems working well. Also you can find Java plugin for Windows XP here:
  http://java.sun.com/getjava/download.html

- **Question:** I am confused:
  In javadoc for public abstract class OutputStream:

  "public abstract void write(int b) throws IOExceptionWrites the specified byte to this output stream."

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

But why in the method we have "int b"?
Was not possible to have public abstract void write(byte b) ?
In the same description later: "The 24 high-order bits of b are ignored."
Why need to use int and ignore 3/4 of that afterwards...

Has it relation to memory management efficiency on 32-bits CPUs and OSs?
What's about 64-bit CPUs like Itanium...
Will SUN change int to 64-bits size?

Or does SUN believe that in future bytes will be 32 bits long? :-)

Answer: Equivalent read() returns an int also. This is to enable a return value of -1 in which is not possible if use byte.
Byte is signed and it has the value range: -128 to 127. And even if the sign is not so important there will not be any possibility to distinguish between -1 and 255!
And last point: byte can not be used directly for mathematical operations, it has to be converted to int anyway.

- **Question:** Why do not distinguish methods based on their return value?
I mean on class names and arguments list? I think, it could be obvious from declaration:

```
void aa(){}
int aa(){}
```

to use it later in a such way:

```
int bbb = aa();
```

compiler could easily distinguish which function must be used here - int, not void...

Answer: Yes, in this situation compiler easily will find out which kind of function must be used. But quite often a method can be called directly - just to do something, not to return some value. The return value is not important here:

```
...
aa();
...
```

In this case compiler can not find out which method must be used. And other programmer reading your program will be confused as well.

- **Question:** I read that Java creates a default constructor for every class if I do not do it myself. Default - without arguments.
I wrote small program where I defined my constructor with two arguments. But in one place I still need an empty constructor (without arguments).
But compiler complains that it can not find it!

Answer: Java creates a default constructor if you have no constructors at all. If you create even one with arguments a default constructor will not be created and it is your responsibility to add it into a program.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** How can I be sure that my package has a unique name?

  Answer: By convention, the first part of the package name is the Internet domain name of the class creator, reversed. Since the Internet domain names are unique you should follow this agreement. Do not use the package name that already match some domain name that you are not owner of. Find free name and register it. You can find cheap registration now on Internet, ~$10/year! For example, http://www.godaddy.com registers for $8.95 only.

- **Question:** Some packages use capital letters in name and some low case? I read they must be low case...

  Answer: Before Java 2 (1.2) the com, org, edu, etc. extensions were capitalized on Java packages. This was changed in Java 2. Now package name is in low case.

- **Question:** Quite often I see that programmers use full name of class in a program. For example like this:

  ```
  java.util.Vector vect = new Vector();
  ```

  I do not like such long lines, they are not so nice!
  Is not it easier to use the "import" in the beginning of program:

  ```
  import java.util.*;
  ```

  and then: Vector vect = new Vector(); ?

  Answer: By using full class name in a program a programmer avoids names collision situation - a case when you could have your own Vector class in your package:

  ```
  com.mypack.utilities.*
  ```

  For example you have:

  ```
  import com.mypack.utilities.*;
  import java.util.*;
  ```

  Compiler will not understand which Vector class you refer in the line below:

  ```
  Vector vect = new Vector();
  ```

  The usage of full class name makes the situation clear.

- **Question:** I use a third party package and I need to have an access to protected methods of one class - the member of that package. I asked developers of the package to change it for me, but they answered that it will take month until the release will be ready (they are occupied with another job now). What can I do? I can not wait with development and need to continue my job.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Exists one legal way and two illegal :-(
Please read license agreement to the package and make sure that you are not going to do it illegally.

1. Just extend the class and you will get access to all protected methods.

2. Take that class, decompile it and change protected to public. Best tool I know is DJ Java Decompiler with graphical user interface.
http://members.fortunecity.com/neshkov/dj.html

3. Add your own class into the package. Write a method that creates an instance of that class where protected method is and call it. Since protected methods are accessible for package members - you can access it now as well from within of your new class.

Please be sure that you are not breaking a license agreement or contact developers and ask a permission to do that.

- **Question:** Is JavaScript the same as Java?

Answer: NO! An Amazingly large number of people, including many web designers, don't understand the difference between Java and JavaScript. Briefly it can be summed up as follows:

Java was developed by Sun Microsystems. Java is a full-fledged object-oriented programming language. It can be used to create standalone applications and applet. Applets are downloaded as separate files to your browser alongside an HTML document, and provide an infinite variety of added functionality to the Web site you are visiting. The displayed results of applets can appear to be embedded in an HTML page (e.g., the scrolling banner message that is so common on Java-enhanced sites), but the Java code arrives as a separate file.

JavaScript on the other hand was developed by Netscape, is a smaller and simpler scripting language that does not create applets or standalone applications. In its most common form today, JavaScript resides inside HTML documents, and can provide levels of interactivity far beyond typically flat HTML pages -- without the need for server-based CGI (Common Gateway Interface) programs.

Some server software, such as Netscape's SuiteSpot, lets web application developers write CGI programs in a server-side version of JavaScript. Both client-side and server-side JavaScript share the same core JavaScript language, but each side deals with different kinds of objects. Client-side objects are predominantly the components of an HTML web page (e.g., forms, text boxes, buttons). Server-side objects are those that facilitate the handling of requests that come from clients, as well as connectivity to databases.

- **Question:** Is Java open source as distributed by Sun, i.e., all the modules including JVMs?
If not, is anyone else doing an open source implementation?

Answer: Java is not open source project. Though you can get the full source code under a Sun license.
http://kaffe.org does open source implementation

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

I read there: "Kaffe is a clean room, open source implementation of a Java virtual machine and class libraries. It is also a fun project that was started by Tim Wilkinson and was made successful by the contributions of numerous people from all over the world.

But Kaffe is not finished yet! You can help by developing new and missing functionality, porting Kaffe to new platforms, and testing your Java applications under Kaffe.

Kaffe mostly complies with JDK 1.1.4, except for a few missing parts.
Parts of it are already JDK 1.2 (Java 2) compatible."

http://gcc.gnu.org/java/ - The GNU Compiler for the Java tm Programming Language

- **Question:** I will be thankful if anyone tells me why JVM is called virtual machine.

  Answer: JVM is called a virtual machine because there is no real hardware which interprets the byte code. If you have done any assembly programming for any microprocessor/microcontroller you will able to understand this. A microprocessor has built-in instruction set to interpret the assembly code. Similarly the JVM is similar to a microprocessor in the sense it has its own instruction set but it implemented in software. That is why it is called a virtual machine!

- **Question:** I read that JDK 1.x has new features...
  I am not quite sure about the stability and performance. Does it mean that 1.x is not suitable for development now?

  Answer: Absolutely not. When you are first starting out on a new project, especially big one it is a good idea to work with the most recent version of something. By the time you are ready to ship, the version of software that you use will have had the bugs shaken out of it. And you can include it into advantages of your product saying that it is built on latest software platform :-)

- **Question:** Do anyone know the difference between java and C#.

  Answer: They are different languages. Java has been around for about five years. C# has not been publicly released yet. One is written by Sun Microsystems, one my Microsoft. They are fairly similar languages with C# having a few extra bits added on to it.
  --
  Phil

  C# bytecodes can be compiled to native exe files just as Java bytecodes can be. But C# is expected to be more closely tied to the Windows operating system and standard interfaces that are part and parcel of Windows. Writing a native compiler that collects all these interfaces and combines them into a unified whole that can run on ANY operating system may require compiling proprietary windows components which Microsoft will make sure is hard to do and against its licensing policies. So you can expect to see native compilers that compile for Windows platforms but not to other operating systems.
  --

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

alankarmisra

- **Question:** I read PHP 4 times faster than JSP. Why then do we need JSP?

  Answer: These tools fill somewhat different niches and seldom directly compete. PHP is good for situations where your page can interact more or less directly with a database, without the need for complex business logic. Its strength is that it can be used to build pages VERY quickly. And, as you note, they run very quickly as well.

  The JSP/Servlet model is more geared toward distributed n-tier applications where there is at least logical, and possibly physical, separation of model, view, and controller functions. It is more complex than PHP, but also more scalable and well-written Java apps may be a great deal more maintainable because of the separation of logical tiers.

  They're both nice tools, and I use both, but I don't think either one is going to kill the other anytime soon.
  --
  Joe

- **Question:** My question is: is JSP as powerful as Servlet? I heard that JSP will eventually compile into Servlet class file.
  One thing can be done by Servlet, can it be done by JSP too? In terms of http.

  Answer: Everything a Servlet does can be done in JSP and vice versa. Good programming practice (you will see some articles over the last year in JavaPro) dictates to combine Servlets and JSP in any significant web application.

  JSP should be mainly HTML (or XML, or WML or whateverML) with little Java inside. Servlets should be Java with few or not at all lines like this:

  ```
  out.println( "<html>" );
  out.printlb( "<body>" );
  ```

  This creates a more or less clean separation between presentation (JSP) and business logic (Servlet).
  Java beans also have a role in this. I strongly recommend the JavaPro articles or whatever text on the MVC model you can find.

  --
  eugene aresteanu

- **Question:** I do not understand the difference between clone () and new (). Why shouldn't I just call new ()?

  Can apply new () to cloned object?

  Answer: The difference is that clone() creates the object that possibly has changed fields (after start up) and this object is willing to be cloned by implementation Cloneable interface.

  New() method can be applied without objects "permission" and create a new instance with fields just initialized and not changed. They will be changed later

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

during runtime. Clone () "mirroring" this modified object.

That's why you NEVER should call new() on cloned object. You can destroy the clone...

- **Question:** Doesn't Java have C's preprocessor functions? At least #define's? If not, the only way to define constants is with 'final'? So... no macros at all?

  Isn't there, then, a separate "preprocessor" utility (at least), if I want to define things like

  #define proc public void
  or so?

  Answer: The answer to that is either "No" or "No, thank God". Code gets read many, many more times than it gets written. You should be constantly making things easy for the developers that have to maintain this after you've finished it (after all, it's probably going to be you).

  That's way Sun pushes its Java coding standards so hard.

  Believe it or not, the only reason that I don't do much C++ work boils down to three things that C++ has that makes its code hideous- the preprocessor, typedef, and operator overloading. All three of those are wholly unnecessary and mung C++ code six ways to Sunday.

  When C# was announced, I was excited because I'd heard its preprocessor was more restrictive. It is, in that it doesn't permit macros. It still lets you do all sorts of whacky "conditional compiling" stuff, though, that makes code unreadable again.
  --
  Cuplan

- **Question:** I'm just starting to learn Java on my own. Should I first learn AWT or should I jump directly into the Swing of things?

  Answer: Will you want to code applets that are easy for anyone to run in their browser? If so, you'll probably have to go with the AWT for now.

  The AWT isn't so bad, but Swing makes a lot of things much easier, so if you want to ship native-code applications or suchlike then I'd go with Swing. I still use the AWT, but I find myself having to code a lot of 'standard' things myself. Swing make things easier but IE doesn't support it.
  --
  Mark

- **Question:** I can't manipulate inodes on my Linux box ... in fact I can't even get real info about a file! Java is a bad hack and is for kids who aren't sharp enough to do C++.

  Answer: Think of Java in the same terms as COBOL and VB, and you've got the right idea. Start thinking of it as a replacement for C++ and you're on the wrong track.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Don't expect this portable language to be a tool for low-level coding with hooks into the OS and hardware internals. It just wasn't designed for that. It's an excellent *applications* language, not a *systems* language like C or assembler.

On the other hand, if any pesky Java programmers tell you that C++ is dead and that Java can do everything C++ does, and more, you may howl with laugher and tell them to eat their JVM.
--
David Ehrens

- **Question:** How do we exchange data between Java and JavaScript and vice-versa?

Answer: Public variable and methods of Java Applet are visible to a HTML document. So using JavaScript you can access the public variables and public functions.

The syntax is:
var some_var = document.appletname.variable_name

With this you will get the value of the variable variable_name in your JavaScript variable some_var.

- **Question:** Constructors and methods: are they the same?

I need a little help here...I have been teaching that constructors are not methods. This is for several reasons, but mainly because JLS says "constructors are not members" and members are "classes, interfaces, fields, and methods."

So, now the rest of the staff is ganging up on me and making life a little nasty. They quote Deitel and Deitel, and Core Java (which references "constructor methods") and who knows how many other books.
The one we are teaching in is loaded with so many errors that even though it calls constructors methods NOBODY will quote it as an authority.

How can so many people call constructors methods if they aren't.

Okay. Are they or aren't they? I holding to the definition that they are not unless it is so common to call them that, that I will have to change. Comments?

Answer: If you go by the JLS (Java Language Specification) and the Java API (and you should) , then no, constructors are not methods. Consider that Class.getMethods () returns an array of Method instances and Class.getConstructors () returns an array of Constructor instances, and Constructor and Method or not interchangeable (one is not derived from the other), but both implement the Member interface. Seems to me that Java is going out of its way to differentiate them.

Besides, the mechanics of constructors are so different from the mechanics of methods, there seems to be no value to considering one a member of the set of the other.
Now, as far as teaching the language goes:
Methods:
+ return types
+ called by name
+ executed multiple times

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Constructors:
+ super(...) or this(...) as first instructor (often implicit)
- everything else

I very much do not like trying to lump the two concepts together, especially in introductory courses. Conceptually they are very, very different things.

A constructor is a unique element (even if there are several). It has the name of the class, its declaration is different, and it doesn't have the same syntax as a method. It plays a unique role. You can call a method from a constructor, but you cannot call a constructor from a method.

I say constructors are distinct from methods, and for students, blurring the distinction will lead to problems.
--
by Chuck McCorvey, Chris Wolfe,
-- Paul Lutus, www.arachnoid.com

- **Question:** I the see method getClass() in java.lang.Object. Do we need it? I know all my classes.

  Answer: Exactly. If you know - you do not need it. But if you do not know then it helps you. For example if you get some object and would like to instantiate it:

```
Object createNewInstOf(Object obj) {
  return obj.getClass().newInstance();
}
```
--
Igor

- **Question:** I know that a default constructor is being defined from line 6 to 9 (see below). But I don't quite understand the syntax: this(blah, blah). Surely I know that "this" refers to the current object. But what on earth does "this (blah, blah, blah)" mean?

  Would you please help explain in what kind of situation we need to use this kind of statement?

```
public class ThreadPool implements Runnable {
  private final int DEFAULT_MINIMUM_SIZE=5;
  private final int DEFAULT_MAXIMUM_SIZE=10;
  private final int DEFAULT_RELEASE_DELAY=10*1000;
  ...
  public ThreadPool(){
    this(DEFAULT_MINIMUM_SIZE, DEFAULT_MAXIMUM_SIZE,
        DEFAULT_RELEASE_DELAY);
  }
  ......
}
```

  Answer: Whenever you encounter the :

  this(blah, blah)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

syntax, it means that another constructor should be called first:

```
public class MyClass{
  MyClass(){
    this(2,2); / / here you make a call to the other constructor
  }

  MyClass(int a, int b){
  }
}
```

The point here can be, that even if the user doesn't know which parameters to pass to MyClass(int, int) , she gets a default constructor which indirectly gives default parameters.

Important notes:
-calling this(...) is very similar to calling super(...) .
-indeed, this(..) may only be used inside a constructor, and may only be placed as the first instruction in a constructor (all like super(...)).

Note that super(...) will call some constructor from the parent class.
--
Arnaud.

- **Question:** Simple **Question:** why constructor doesn't work in following example?

```
class Start {
  public void Start() {
    System.out.println("constructor START");
  }
}

public class Test {
  public static void main(String[] args) {
    Start s = new Start();
  }
}
```

Answer: Because you have included the return-type 'void' in the method declaration, it becomes a normal method, which just happens to have the same name as the class - so it won't get used as a constructor. Remove the 'void' and it should work.
--
Vince Bowdren

P.S. by John: If you do not specifically define any constructors, the compiler inserts an invisible zero parameter constructor "behind the scenes". Often this is of only theoretical importance, but the important qualification is that you only get a default zero parameter constructors if you do not create any of your own.

Your program used this zero parameter constructor and you saw nothing...

- **Question:** Why we can not declare constructor as final?

Answer: The keyword final when dealing with methods means the method cannot

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

be overridden.

Because constructors are never inherited and so will never have the opportunity to be overridden, final would have no meaning to a constructor.

- **Question:** In Java, does exist a function like sprintf in C?

  Answer: http://www.efd.lth.se/~d93hb/java/printf/index.html a free Java version of fprintf(), printf() and sprintf() - hb.format package

- **Question:** If I declare an array of an objects, say Dogs, is that memory taken when I create the array or when I create the objects in the array when I declare this array:

  Dog[] dog = new Dog[100];

  or does it take the memory when I actually create the Dogs in the array e.g.:

  ```
  for(int i = 0;i<dog.length;i++)dog[i] = new Dog();
  ```

  Answer: The statement above is actually two-fold. It is the declaration and initialization of the array. Dog[] dog is the declaration, and all this does is declare a variable of type Dog[], currently pointing to null.

  You then initialize the array with new Dog[100], which will create 100 elements in the array, all of them referencing null.

  It is important to realize that the elements of an array are not actually objects, they only reference objects which exist elsewhere in memory. When you actually create the Dog objects with new Dog(), these objects are created somewhere in memory and the elements in the array now point to these objects.

  Pedant point:

  Nothing ever points to null. It is a constant that represents the value of a reference variable that is not a pointer to some object new Dog[100] creates an array of 100 null Dog references.

- **Question:** How do I return more than one value using the return command?

  Answer: You could make a new object/class that contains these two values and return it. For example:

  Define an object like this:

  ```
  class MyObj {
    public int myInt;
    public double myDouble;
  }
  ```

  Then, in your method create one of these, set the corresponding values, and return it.

  ```
  MyObj yourMethod() {
    MyObj obj = new MyObj()
  ```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  obj.myInt = 20;
  obj.myDouble = 1.0003
  return obj;
}
```

- **Question:** How do I use object serialization for an object that has other objects as data member? Do both the class need to implement serialize?
How about static data?

```
class A{
}

class B{
  public A a;
}
```

Answer: Both the object and all the object references it contains need to belong to classes that implement Serializable.

Static and transient fields are not serialized. For more, see,
http://java.sun.com/docs/books/tutorial/essential/io/serialization.html

- **Question:** I recently learned a bit about "inner classes" but this seems to be different...

I'm a bit new to Java programming so bear with me. My employer bought a package of java graphics library programs to support some chart applets we want to create. We have the source code. I'm trying to create a jar file with all the files I need to run the applet. When I currently run the applet, the browser java tool says that it can't find "TextComponent$1.class".

I recently learned a bit about "inner classes" but this seems to be different. The "TextComponent.java" file does contain some inner classes, but not a class called "1". I'm confused.

Is this an inner class? Or is it something else. Any help would be appreciated. Thanks...

Answer: The TextComponent$1.class is the first anonymous class defined in TextComponent.java. Since nested (inner) classes are compiled to their own .class file, they needed unique names. The javac compiler is just creating a unique file name for an anonymous nested class.

- **Question:** Hi there, does anybody know a good source of design patterns written in JAVA?

Answer: A pretty good (free to download) book.
http://www.patterndepot.com/put/8/JavaPatterns.htm

- **Question:** What's the difference between the two: System.err. and System.out? When should we use System.err?

Answer 1: System.out leads the output to the standard output stream (normally

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

mapped to your console screen), System.err leads the output to the standard error stream (by default the console, too). the standard output should be used for regular program output, the standard error for error messages. If you start your console program regularly both message types will appear on your screen.

But you may redirect both streams to different destinations (e.g. files), e.g. if you want to create an error log file where you don't want to be the regular output in.

On an UNIX you may redirect the output as follows:

java yourprog.class >output.log 2>error.log

this causes your regular output (using System.out) to be stored in output.log and your error messages (using System.err) to be stored in error.log

- **Question:** What's the difference between the two: System.err. and System.out? When should we use System.err?

  Answer 2: System.err is a "special" pipe that usually is directed to the standard console. You can redirect the System.out with the normal pipe control (| or >), but System.err no. If you want to put both the "normal" output and the "error" output to a file you must use the special redirect 2>.

  This allow you to send normal messages into a file or in the /null black hole, but still receive the error messages on the console.

- **Question:** What is the essential difference between an abstract class and an interface? What dictates the choice of one over the other?

  Answer: You can only extend one class (abstract or not) whereas you can always implement one or more interfaces. Interfaces are Java's way to support multiple inheritances.

  Does anyone know how could I get the size of an Enumeration object? The API for Enumeration only contains getNext () and next ().

  Answer 1: You can't. Theoretically, some classes that implement Enumeration may also provide some way to get a size, but you'd have to know about the more specific run-time type and cast to it... and none of the standard java.util Collections classes nor Vector or such provides these methods in their Enumeration implementations.

- **Question:** What is the essential difference between an abstract class and an interface? What dictates the choice of one over the other?

  Answer 2: You can make your own class like this:

```
import java.util.*;

public class MyEnumeration{
   int size;
   int index = 0;
   Enumeration e;
   public MyEnumeration(Vector v){
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    size = v.size();
    e = v.elements();
    index = 0;
  }
  public boolean hasMoreElements(){
    return e.hasMoreElements();
  }

  public Object nextElement(){
    index++;
    return e.nextElement();
  }

  public int size(){
    return size;
  }

  public int getIndex(){
    return index;
  }
}
```

--
by Nicolas Delbing and Victor Vishnyakov

- **Question:** Is there a way to provide values for a Vector in the source code, analogous to array initializers?

  Answer: The Vector class constuctors take no arguments other than Collection (since JDK 1.2), which is abstract, and since a Vector is a structure whose size can change dynamically, it's contents can only be initialized through member methods.
  --
  Mike Lundy

- **Question:** How would I add a help file to a java application?
  Would it have to be platform specific, or is there a Java API for making help files? If so, what is it?

  Answer: See JavaHelp at http://www.javasoft.com/products/javahelp/
  you create HTML pages for the main text, and add some XML files for a hierarchical table of contents and a map from TOC tags to relative URLs giving document locations.

- **Question:** What is a Just-In-Time (JIT) compiler?

  Answer: It is a JVM that compiles Java instructions (called bytecode) into native machine instructions at run time and then uses this compiled native code when the corresponding Java code is needed. This eliminates the constant overhead of interpretation which tradition first generation JVM's used.
  --
  Dave Lee

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** Is there a collection object like the hashmap or hashtable that stores values in an ordered path? Vector does this but I need the key/value functionality. Hashmaps do not guarantee the order of the objects.

  Answer: Take a look at java.util.TreeMap.
  Red-Black tree based implementation of the SortedMap interface. This class guarantees that the map will be in ascending key order, sorted according to the natural order for the key's class (see Comparable), or by the comparator provided at creation time, depending on which constructor is used.

  Note that this implementation is not synchronized. If multiple threads access a map concurrently, and at least one of the threads modifies the map structurally, it must be synchronized externally.

- **Question:** Most people asked why there is an error, but my question is why this is NOT an error...

  Please take a look:

  r is a number and s is a character, why can I put them together to make a comparison without compilation error? Could somebody tell me... thank you

  ```
  double r = 34.5;

  char s = 'c';
  if (r > s) {
     System.out.println("r > s");
  } else {
     System.out.println("r < s");
  }
  ```

  Answer: Yes, char is indeed a 16-bit value. However, the actual answer is in the Java Language Specification, section 5.6.2, which is at the following URL: http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#170983

  In summary, the char is automatically promoted to a double. No explicit cast is necessary since the language rules say that it gets "promoted" to a double
  --
  by John O'Conner

- **Question:** == and equals ()... These two still make me confuse a lot of time. Can somebody give me some thumb rule or explain it to me?

  Answer: When you use == with a primitive -int, double, char...you are checking that the values are identical. But if you use == with an object, you are checking that the 2 objects are stored at the same address. In other words the references pointing to the same object...

  Method equals () is different.
  It is the same as ==, if it isn't overridden by the object class.

  Many classes override the method equals (). In this case this method will check that

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

content of the object is the same or not, not addresses.

- **Question:** Why do I get message like “wrong magic number” when I am trying to run applet? What is magic number?

  Answer: The first thing a JVM does when it loads a class is check that the first four bytes are (in hex) CA FE BA BE. This is the "magic number" and that’s why you are getting that error, you are trying to load a file that isn’t a class and so the class loader in the JVM is throwing out that exception.

  Make sure you transfer the class files to site in binary mode, rather than text or ASCII mode.

  An error from the browser saying "cannot start applet ... bad magic number" usually means that one of the class files on the server is corrupted. '

  Replace your class binary files on the web server; clean up the cache of your browser, and reload your applet.

- **Question:** In java, I found a lot of methods, which let you enter a line (read (), readLine () etc). They all wait until return is pressed, and then start providing you the information.

  Does anyone know if there is a read method available whit the desired behavior, i.e. which doesn't wait for return being pressed?

  Answer: Java does not provide it, the terminal itself waits until return is pressed before sending the entered line to Java.

  You need to use some platform specific mechanism to change the terminal settings.

- **Question:** Can I pass a function as a parameter in Java? If so, how?

  Answer: No, you cannot. But you can pass an object with method and then just use this method like this:

  ```
  myFunction (object); // object has method useMyMethod(){ do smth here...}
  ..
  ..
  somewhere in another place where use this function..
  object.useMyMethod();
  ```

- **Question:** What is difference capacity() and size() methods for vector?

  What is difference between

  public final int capacity()
  Returns the current capacity of this vector.

  and

  public final int size()
  Returns the number of components in this vector.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: please read this method: ensureCapacity()

public final synchronized void ensureCapacity(int minCapacity)

Increases the capacity of this vector, if necessary, to ensure that it can hold at least the number of components specified by the minimum capacity argument.

Parameters:
minCapacity - the desired minimum capacity.

- **Question:** Can I use try/catch block to control program flow? Will it affect program performance?

Answer: Try/catch block itself does not add much overhead to your program. Handling of thrown exception can be expensive.
Use try/catch block only for error handling, not to control program flow. Especially try to avoid placing within loops.

- **Question:** I want to create a logging system that handles the user, database, application server logging, etc.

What are options?

Answer: Sun has introduced a new API for logging in JDK 1.4. It is in the package java.util.logging.

If you use a JDK prior to 1.4, use alternative Log4j from the Apache project (jakarta.apache.org/log4j).

It is an open source logging API. Very easy to use!

If you want to use something else on JDK 1.2 and 1.3 but similar to the final java.util.logging API in 1.4, use:

http://javalogging.sourceforge.net

- **Question:** I have a method that names something like this - public String doSmth()...

I do not need any Object for this method. Can I make it static? Is it good idea to make all methods that do not use Objects static?

Answer: yes, I think so. It is bad idea to create an instance of the class just to use one method without needing the instance itself.

Wasting of memory and CPU performance for further garbage collecting of the Object that has not been used :-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** What're the differences between classpath and import in the java application?

  Do I need to have the package in the classpath first before importing in a java application or need not?

  Answer: Classpath is an environment variable of your OS, you have to set it (or better: Java sets it) to tell Java where to search for classes.

  You use import in a program to let Java search for the specified classes within the classpath. This implies that the package must be in the classpath.
  --
  Stephan Effelsberg

- **Question:** What is difference between Iterator and Enumeration?

  First of all Java FAQ Team wish you HAPPY NEW YEAR!!! and then

  Answer: from
  http://java.sun.com/docs/books/tutorial/collections/interfaces/collection.html

  The object returned by the iterator method deserves special mention. It is an Iterator, which is very similar to an Enumeration, but differs in two respects:

  Iterator allows the caller to remove elements from the underlying collection during the iteration with well-defined semantics.

  Method names have been improved.

  The first point is important: There was no safe way to remove elements from a collection while traversing it with an Enumeration. The semantics of this operation were ill-defined, and differed from implementation to implementation.

  The Iterator interface is shown below:

```
public interface Iterator {
  boolean hasNext();
  Object next();
  void remove(); // Optional
}
```

  The hasNext method is identical in function to Enumeration.hasMoreElements, and the next method is identical in function to Enumeration.nextElement. The remove method removes from the underlying Collection the last element that was returned by next. The remove method may be called only once per call to next, and throws an exception if this condition is violated. Note that Iterator.remove is the only safe way to modify a collection during iteration; the behavior is unspecified if the underlying collection is modified in any other way while the iteration is in progress.

  The following snippet shows you how to use an Iterator to filter a Collection, that is, to traverse the collection, removing every element that does not satisfy some condition:

```
static void filter(Collection c) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
   for (Iterator i = c.iterator(); i.hasNext(); )
      if (!cond(i.next()))
      i.remove();
}
```

Two things should be kept in mind when looking at this simple piece of code:

The code is polymorphic: it works for any Collection that supports element removal, regardless of implementation. That's how easy it is to write a polymorphic algorithm under the collections framework!

It would have been impossible to write this using Enumeration instead of Iterator, because there's no safe way to remove an element from a collection while traversing it with an Enumeration.

- **Question:** How can I find the first dimension length of the 2-dimenstions array? I use the array [].length but it does not work, how can I solve this problem?

  Answer: Java doesn't really have "multidimensional arrays", only arrays of arrays. So try: array[0].length and you will get this dimension.

- **Question:** I guess what I'm asking is "Is java.util.Hashtable thread safe?"

  It's been a while since I've used hashtables for anything significant, but I seem to recall the get() and put() methods being synchronized.

  The JavaDocs don't reflect this. They simply say that the class Hashtable is synchronized. What can I assume? If several threads access the hashtable at the same time (assuming they are not modifying the same entry), the operations will succeed, right? I guess what I'm asking is "Is java.util.Hashtable thread safe?"

  Answer: That is right! It is recommendable, if you have questions like these, always look at source for the API, it's freely available.

- **Question:** I try to copy an object of my own using the clone() method from java.lang.Object, but this is a protected method so I can't use it. Is there some other way to get my objective of duplicating an arbitrary object?

  Answer: If you want to clone your object, you need to make it cloneable. To achieve this, you need to do two things:

  1. implement the interface Cloneable
  2. override the method clone(), so that it
     a. becomes public
     b. calls super.clone()
     c. if necessary, clones any members, or
     d. if a member can't be cloned, creates a new instance.

  Simple example:

```
public MyClass implements Cloneable {
   int someNumber;
   String someString;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public Object clone() {
  // primitives and Strings are no
  // problem
    return super.clone();
 }
}
```

In this case the method clone() of the class MyClass returns a new instance of MyClass, where all members have exactly the same value. That means the object reference 'someString' points to the same object. This is called a shallow copy. In many cases this is no problem. Strings are immutable and you do not need a new copy. But if you need new copies of members, you have to do it in the clone() method. Here is another simple example:

```
public class SomeMember implements Cloneable {
  long someLong;
  public Object clone() {
    return super.clone();
  }
}

public AnotherClass extends MyClass {
   SomeMember someMember;
   public Object clone() {
      AnotherClass ac = (AnotherClass)(super.clone());
      if (someMember != null) {
         ac.someMember = (SomeMember)(someMember.clone());
      }
      return ac;
   }
}
```

Note that the class AnotherClass, that extends MyClass, automatically becomes Cloneable, because MyClass is Cloneable.

Also note, that super.clone() always returns an Object of the type of the actual object, although the superclass doesn't know anything about that sub class. The reason is, that Object.clone() is a native method, which just allocates new memory for the new object and copies the bytes to that memory. Native code has it's own ways of finding out which type to return ;-)
--
Karl Schmidt

- **Question:** I was just wondering about the usefulness of Interfaces...
I was just wondering about the usefulness of Interfaces. I was under the impression that interfaces could be used to perform multiple inheritance. But an interface only declares a method - in a very abstract way.

A class that implements an interface needs to define its own implementation of a certain method. What is the use of having an interface when nothing is being gained...?

Answer: If two classes implement the same interface, you can get a reference to the interface instead of the effective class without bother what class are you

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

managing.

This is very useful in RMI (for example) or in any condition when you have to take an object without knowing exactly his class, but only the interface that it implement.

For example:

public void recurseList(List l)

the generic List ensure that you can use every List for this method (ArrayList, AbstractList, Vector...), so your calling method can be:

ArrayList l = new ArrayList(); or
Vector l = new Vector();
recurseList(l);
Without any problem.
--
by Davide Bianchi

- **Question:** I got a problem with an array/vector... I got a problem with an array/vector.

my class contains a member:
static Vector quad[][];
....

in my constructor I got:

```
Vector quad[][] = new Vector[row][col];
for (int i = 0; i < row; i++){
  for (int j = 0; j < col; j++){
    quad[i][j] = new Vector (0,1);
  }
}
```

// row and col are int between (10..50) -- it's a big I know, but that might not be the problem.

My PROBLEM that (and I don't know what to do, really) I can't access quad[x][y] outside of the constructor!!!! Within the constructor I've got full access on quad[x][x]. Java (1.2) returns a NullPointerException on any method except within the constructor!!!

What's my fault!???

- **Question:** What is Java Space technology?

Answer: The dominant model of computation in distributed computing is the Client\Server model. This model is based on the assumption that local procedure calls are the same as remote procedure calls.

Javaspace technology is based on a new model of computation called the tuplespace model and aims to overcome some of the key difficulties related to Client\Server communication.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Javaspaces overcome the problems of synchronization, latency and partial failure, inherent in distributed systems, by providing loosely coupled interactions between the components of distributed systems.

Javaspace processes communicate through a space, not than directly. Communication between processes on different physical machines is asynchronous and free from the main limitation of the traditional client/server model, when client/server communication requires simultaneous presence on network both parts - client and server.

JavaSpace technology. Sender and receiver in Java Space don't need to be synchronized and can interact when network is available...

In a distributed application, JavaSpaces technology acts as a virtual space between providers and requestors of network resources or objects. This allows participants in a distributed solution to exchange tasks, requests, and information in the form of Java technology-based objects.

Question: How is JavaSpaces technology different from a database?

Answer: Databases require a tremendous amount of overhead in structured programming, screen design, type matching, and indexing to achieve their results. Clients post specific questions to the server, and receive a specific answer. JavaSpaces technology provides a more loosely coupled repository of information. The identity of a client or a server is no longer relevant; data packets are treated just like any other object posted to the space as an anonymous service.

More importantly, JavaSpaces technology can host more than just data. A space can find, match and reference objects by both type and value, meaning they can store objects as information or behavior. This means any object-based program, device, or user profile can join a JavaSpace system. This capacity greatly enhances the ability to coordinate different functions and processes in a network environment.

Java Space technology is ready for usage product although it is still under development. One example of Java Space based product here:

http://j-spaces.com/

This newsletter is based on:

"JAVASPACES: A CRITIICAL ANALYSIS OFF CONCEPTS & TECHNIQUES"

http://www.dcs.shef.ac.uk/teaching/eproj/ug2001/pdf/u8sm.pdf

"JavaSpacesTM FAQ"

http://java.sun.com/products/javaspaces/faqs/jsfaq.htm

Question: What is meant by persistence? What is relation of persistence to high availability?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: Persistence indicates that a collection of data remains intact even if its source is no longer attached to the network. The objects in JavaSpaces for example, remain available to other users even if the source has temporarily disconnected from the network. This feature is critical to keep distributed systems highly available.

Question: I heard many times from Java lectors: "If you are interested how new feature was implemented in JDK please check the source code...". I tried to find the source code but have not succeeded.

Of course, I can decompile class files but could be good to have "original" source code from SUN, not from hacking tools :-)

Answer: I found the SUN's page and what was there:

"Sun is making the source code to the Java 2 platform available to the developer community. This offering is intended to further propel innovation of the Java platform and provide developers with easier access to Java platform sources while still maintaining compatibility.

Do You Need The Source Code?

In most cases, you do not need the source. In fact, most developers will find all the basic resources and tools they need in the Java 2 Software Development Kit (formerly known as the JDK 1.2).

The source code to the Java 2 platform can be invaluable if you are doing any of the following:

conducting research work
debugging possible platform bugs
adding enhancements which require access to platform internals

But building, porting, and testing the Java 2 platform source code requires a high level of technical expertise. "

P.S. About last sentence: I am pretty sure that all readers of my newsletter have it.

Question: I am getting "OutOfMemoryError". How do I prevent such memory leaks?

Answer: It is not always a memory leak that causes "OutOfMemoryError".

A memory leak is a situation when some application, for example JVM allocates some amount of memory and then loses

pointers to that area. So, the allocated area becomes a "black box" which you can not access and you need a key to open it.

It is very rare situation when some Java code causes memory leaks. They are reported and can be found at "Bug parade" on SUN's web site.

"OutOfMemoryError" often caused by bad design. It can be for example:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

1. Your program trying to process too much data. Increase the amount of memory or decrease the amount of data to be processed. Actually you can do it in a wise way: control how much memory left and if it less than, let say 20 %, decrease the processing speed. It was described in one of our tips while ago...

2. Your program holding too many objects and then program is running out of memory.

You need to use close methods if you do need them. You can even explicitly set a reference to null to speed up garbage collection.

3. You use too many objects with finalizers... They consume your memory before you get any chance to run them...

Question: I have an application which can be run using 64 bit JNI libraries or 32 bit libraries depending on the command line parameters. I want to know if there is any way for me to find out if the JVM is running in 64 bit mode (the -d64 parameter was

sent to it) or 32 bit mode.

Answer:

Run this small program and you will get a long list of System properties...

```
public class ShowSystemProperties {
  public static void main(String[] args) {
    System.getProperties().list(System.out);
  }
}
```

One of lines will be

sun.arch.data.model=32

or

sun.arch.data.model=64

Just search for it and you will know which version you are running on.

Question: What is the maximum number of dimensions that a Java array can take?

Answer: It is not limited by Java Language Specification, but limited by implementation...

For example, for java.lang.reflect.Array newInstance method I found this:

"The number of dimensions of the new array must not exceed the number of array dimensions supported by the implementation (typically 255)."

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Question: Is there any way to keep an object from being collected by Garbage Collector?

Answer: 8-)

Yes, of course! Just keep a reference to an object and it will not be garbage collected (according to definition of GC)!

Question: Reading Java's API version 1.41 I noticed that many classes contain now valuable information about thread safety. Are all classes having such useful info?

Answer: That is almost right. SUN added descriptions to many classes. For example see bug No 4264153

(http://developer.java.sun.com/developer/bugParade/bugs/4264153.html)

I am not quite sure that all classes have a description. At least I have not found any information that SUN updated ALL descriptions...

My recommendation is to read API... If there is no information on thread safety for particular class to assume that class is not safe. It will help to avoid strange runtime errors which are difficult to debug.

Question: Do you know about "Full Speed Debugging" and "HotSwap" Class File Replacement in Java Debugger since v.1.4.1? Part 1.

Answer:

Full Speed Debugging

The Java HotSpot[TM] virtual machine now uses "full-speed debugging". In the previous version of HotSpot, when debugging was enabled, the program executed using only the interpreter. Now, the full performance advantage of HotSpot Technology is available to programs running with debugging enabled. The improved performance allows long running programs to be more easily debugged. It also allows testing to proceed at full speed and the launch of a debugger to occur on an exception:

-
Xrunjdwp:transport=dt_socket,server=y,onuncaught=y,launch=myDebuggerLaunch Script

For details on launch "onthrow" and launch "onuncaught", see Connection and Invocation Details . Setting a breakpoint only inhibits compilation (full speed execution) for the method containing the breakpoint. Execution reverts to the interpreter during some debug operations such as single-stepping and when method entry/exit or watch points are set.

As of version 1.4.1, full-speed debugging is available with both the Java HotSpot Client Vm (default) and Java HotSpot Server VM (invoked with the -server command-line flag). In the Java 2 SDK v1.4.0, full-speed debugging was available only with the Java HotSpot Client VM.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

This tip is based on release info for the version 1.4.1 family of the Java 2 SDK.

Question: Do you know about "Full Speed Debugging" and "HotSwap" Class File Replacement in Java Debugger since v.1.4.1? Part 2.

"HotSwap" Class File Replacement.

This new feature encapsulates the ability to substitute modified code in a running application through the debugger APIs. For example, one can recompile a single class and replace the old instance with the new instance.

This change was made to address these issues:

Tool (IDE) vendors want the ability to do fix-and-continue debugging. That is, while debugging, identify a problem, fix it, and continue debugging with fixed code.

Organizations deploying long running servers wish to be able to fix bugs without taking down the server.

HotSwap adds functionality to the Java TM Platform Debugger Architecture (JPDA) to allow a class to be updated while under the control of a debugger. The two central components of this functionality are RedefineClasses which replaces the class definitions and PopFrame which pops frames off the stack allowing a method which has been redefined to be re-executed.

In the reference implementation, this functionality is implemented at the Java Virtual Machine Debug Interface (JVMDI) layer and made available through the higher layers of JPDA - the Java Debug Wire Protocol (JDWP) and the Java Debug Interface (JDI).

This tip is based on release info for the version 1.4.1 family of the Java 2 SDK.

Question: I see that in Java a lot of Class loaders: bootstrap, extension, system, context and even user defined! It is difficult to understand Class loader hierarchy in Java..

Answer: It was much better in Java 1.1 time :-)

There were no relations between system class loader (responsible for loading in the Java runtime, the application, and classes and resources in the application's CLASSPATH) and applet class loader (loaded classes over networks).

Since Java 1.2 we have three types of class loaders:

1. Class loaders created automatically by the JVM
2. Program defined class loaders
3. Context class loaders.

There are three Class loaders in first group:

bootstrap class loader - loads classes from ../jre/lib/rt.jar
It is the "root" in the class loader hierarchy.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

extensions class loader - loads classes from ../jre/lib/ext/*.jar
system class loader - it is responsible for loading in the application, as well as for loading classes and resources in the application's CLASSPATH.

Second group includes system class loader - parent class loader by default additional parent class loader can be specified explicitly

Third group is context class loader. A thread's context class loader is, by default, set to the context class loader of the thread's parent. The hierarchy of threads is rooted at the primordial thread (the one that runs the program). The context class loader of the primordial thread is set to the class loader that loaded the application. So unless you explicitly change the thread's context class loader, its context class loader will be the application's class loader. That is, the context class loader can load the classes that the application can load.

To change a thread's context class loader, you use Thread.setContextClassLoader().

Question: I am getting "StackOverflowError" error...

Where should I look at?

Answer: "StackOverflowError" usually caused by an infinite recursion.

Ł **Question:** I got a problem with an array/vector...

my class contains a member:
static Vector quad[][];
....
in my constructor I got:

```
Vector quad[][] = new Vector[row][col];
for (int i = 0; i < row; i++){
   for (int j = 0; j < col; j++){
      quad[i][j] = new Vector (0,1);
   }
}
```

// row and col are int between (10..50) -- it's a big I know, but that might not be the problem

My PROBLEM (and I don't know what to do, really), I can't access quad[x][y] outside of the constructor!!!! Within the constructor I've got full access on quad[x][x]. Java (1.2) returns a NullPointerException on any method except within the constructor!!!

What's my fault!???

Answer: I guess you shouldn't write Vector here:

Vector quad[][] = new Vector[row][col];

so, the correct variant may be:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

quad[][] = new Vector[row][col];

I guess You just overridden your static variable with one defined in your constructor: Vector quad[][].

Thus, you're initializing NOT your class-scope static variable but your constructor-scope quad. It's not reachable outside the constructor. And as for static quad, it has never been initialized! And a first reference to it causes NullPointerException. I guess. I hope I'm right :)
--
Xao Rom

- **Question:** I know that Java file should have the same name as public class in the file.

  But what are rules for files that have no public classes?

  Answer: No rules are defined for such case! It is up to you how to name it. Please check my example:

```
// ******* ABCDEFG.java file ********************
class G {
  public static void main(String[] args) {
    System.out.println("This is class G!");
  }
}

class Z {
  public static void main(String[] args) {
    System.out.println("This is another class Z!");
  }
}
class AB { }
class CD { }
class EF { }
```

  After compilation you will get AB, CD, EF, G and Z classes. You can run G and Z (they have main method), but not AB, CD, EF.

  Try it:
  java G

  or

  java Z

- **Question:** I mainly use C++ but I have a program to write and java seems to be the right tool for this job. That said my java skills are a bit rusty. I downloaded a free IDE and am ready to go but I'm not a fan of putting my method implementations in the class definition.
  Is there a way to separate the implementation from the class definition? Like in c++ there are .h files and .cpp files.

  Answer: You can use an interface and an implementation class, if you like.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

The interface has just the method signatures; the class has their implementations. You can then declare variables having the type of the interface, and assign objects of the implementation class created with "new", or by a factory method.

Java is very different from C++ in that the interface of a class is defined by a compiled view of that class, not by textual inclusion of source for a separate interface description. Therefore the concept of splitting interface from implementation does have the same meaning in Java as in C++.

In Java, you can separate the description of pure interfaces (or abstract classes) from concrete classes that implement (or complete) them. This is a useful technique - compare java.util.Collection and java.util.List to java.util.ArrayList and java.util.LinkedList.

Also, it is traditional to define the return type from factory methods and other such producers to be pure interfaces to allow the factory or other class to decide which implementation to give you - this is used extensively in java.net and java.sql packages.
--
Phil Hanna (Author of JSP: The Complete Reference http://www.philhanna.com), Chuck

- **Question:** How do we exchange data between Java and JavaScript and vice-versa?

  Answer: Public variable and methods of Java Applet are visible to a HTML document. So using JavaScript you can access the public variables and public functions.

  The syntax is:
  var some_var = document.appletname.variable_name

  With this you will get the value of the variable variable_name in your JavaScript variable some_var.

- **Question:** Constructors and methods: are they the same?

  I need a little help here...I have been teaching that constructors are not methods. This is for several reasons, but mainly because JLS says "constructors are not members" and members are "classes, interfaces, fields, and methods."

  So, now the rest of the staff is ganging up on me and making life a little nasty. They quote Deitel and Deitel, and Core Java (which references "constructor methods") and who knows how many other books.

  The one we are teaching in is loaded with so many errors that even though it calls constructors methods NOBODY will quote it as an authority.
  How can so many people call constructors methods if they aren't.

  Okay. Are they or aren't they? I holding to the definition that they are not unless it is so common to call them that, that I will have to change. Comments?

  Answer: If you go by the JLS (Java Language Specification) and the Java API (and you should), then no, constructors are not methods. Consider that

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Class.getMethods() returns an array of Method instances and Class.getConstructors() returns an array of Constructor instances, and Constructor and Method or not interchangeable (one is not derived from the other), but both implement the Member interface. Seems to me that Java is going out of its way to differentiate them.

Besides, the mechanics of constructors are so different from the mechanics of methods, there seems to be no value to considering one a member of the set of the other.

Now, as far as teaching the language goes:
Methods:
+ return types
+ called by name
+ executed multiple times

Constructors:
+ super(...) or this(...) as first instructor (often implicit)
- everything else

I very much do not like trying to lump the two concepts together, especially in introductory courses. Conceptually they are very, very different things.

A constructor is a unique element (even if there are several). It has the name of the class, its declaration is different, and it doesn't have the same syntax as a method. It plays a unique role. You can call a method from a constructor, but you cannot call a constructor from a method.

I say constructors are distinct from methods, and for students, blurring the distinction will lead to problems.
--
by Chuck McCorvey, Chris Wolfe,
-- Paul Lutus, www.arachnoid.com

- **Question:** I am wondering if a constructor has return type...

  Answer: It has no return type, not even void...
  But speaking more generally we can assume that implicit return type is a class type itself.

- **Question:** Constructors are similar to methods... I know that they do not return any value. Could I say that they have void type (no return type)?

  Answer: Not. Constructors are not methods and they are different because they:
  * always have the same names as the class name
  * have no return type, even void!
  * are not inherited and that's why (by the way) you can declare them final.

- **Question:** Ok, I know that one cannot put primitive types into a hashmap (only objects or references to them) and I know how to deal with that (write some kind of wrapper class).

  What I'm interested in is: 'Why is that?' Why can I not put a primitive type into a hashmap? Something to do with this 'heap' thing... Right?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: HashMap requires a key or a value to be assignment- compatible with java.lang.Object (i.e. to be an object or an array). Primitive types aren't ones.

You can use wrappers like java.lang.Integer for this purpose. All container classes require some basic operations to be defined for all of its contained objects in order for it to organize the data.

For example, a hashmap requires a hashing method hashCode (). A primitive type doesn't have any methods associated with it.

The distinction between primitive types (int) and their wrapper classes (Integer) is made purely for optimization purposes. For example, we wouldn't want to have to instantiate actual objects in an array instead of just allocating a block of memory for an array of integers.
--
Gary

Ł **Question:** I propose that Java should allow multiple inheritances if...

Everyone knows the potential problem with multiple inheritances is when you run into the problem of having two instances of a grand parent super class.

For example:

```
class A extends D { int i; }
class B extends D { int i; }
class C extends A,B { }
```

Potentially, you could have two copies of D for each instance of C.
However, I propose that Java should allow multiple inheritance if there are no instance variables associated with the abstracts that the base class is extending.

```
abstract class A {  public setX(); public setY(); public setAll() { setX (); setY(); }
abstract class B {  public setC(); public setD(); public setBoth()
{ setC(); setD(); }

class C extends A,B { }
```

You won't have two instances of some grandfather class, since A and B doesn't have instances variables.

I hope the next versions of Java explore this issue.

Answer: It does. They're called interfaces:

```
interface A {  public void setX(); public void setY(); public void setAll(); }
interface B {  public void setC(); public void setD(); public void setBoth(); }

interface C extends A,B { };
public abstract class D implements C {
}
--
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

jim

- **Question:** Can I access a private variable of one Object from another Object?

Answer: Yes, if this object of the same class. You can access private field from static method of the class through an instance of the same class.

Check one example below:

```
public class Example {
  private int privateVar;
  Example(int a) {
    privateVar = a;
    System.out.println("private privateVar = " + privateVar);
  }

  public static void main(String[] args) {
    Example ex = new Example(5);
    System.out.println(ex.privateVar);
  }
}
```

--
AP (JA)

- **Question:** Is there a way to know from class X which class called the method foo()?

If class A and class B are calling a method foo() on class X, is there a way to know from class X which class called the method foo() (they can be either A or B). I know that this can be done by capturing the stack trace and examining it, but that solution looks expensive as I have to create a new Throwable object every time and capture stack trace (And I do this quite frequently).

Is there any other elegant solution to do this, any help and directions appreciated.

Answer: Pass a reference to the class to the foo() method.

```
foo(Object x){
    System.out.println(x.getClass());
}
```

should do it.

- **Question:** Why does this simple application never exit?

```
public class UIQuitTest {
   public static void main (String[] args) {
      java.awt.Frame f = new java.awt.Frame();
      f.dispose();
      f = null;
  } // end of main ()
}
```

The application above never quits, is it a bug or a (mis)feature? Win98, JRE 1.3.0

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: By creating an AWT object, you now have started the AWT thread. In order to end the application now, you have to do a System.exit (0) that will kill all non-daemon threads, including the AWT thread.

- **Question:** Is it possible to stop an object from being created during construction?

For example if an error occurs inside the constructor (e.g. the parameters pass in were invalid) and I wanted to stop an object being created would it be possible to return null rather than a reference to a new object. (I know the term return is technically correct in this case but you know what I mean).

Basically, is it possible to cancel object creation?

Answer: Yes, have the constructor throw an exception. Formally, an object _will_ be created (since the constructor is a method invoked after the actual method creation), but nothing useful will be returned to the program, and the dead object will be later reclaimed by Garbage Collector.

But the clean way is as another reply suggests, that you leave calls to the constructor to a static factory method which can check the parameters and return null when needed.

Note that a constructor - or any method in general - throwing an exception will not "return null", but will leave the "assign target" as it was.
--
Tor Iver Wilhelmsen

- **Question:** What does mean "volatile"?
For the past couple of hours, I've seen quite a few set of codes that has the _volatile_ keyword.

E.g.
private volatile somethingsomething....

What does it mean?

Answer: See JLS 2nd Edition, which just came out last year, still mentions it.

http://java.sun.com/docs/books/jls/second_edition/html/classes.doc.html#36930

"A field may be declared volatile, in which case a thread must reconcile its working copy of the field with the master copy every time it accesses the variable. Moreover, operations on the master copies of one or more volatile variables on behalf of a thread are performed by the main memory in exactly the order that the thread requested."

Seems just like the idea in C++ and appears still to me present in the language.

Synchronization certainly has it place in many applications, that doesn't mean that volatile is not longer used or part of the language.
--
Paul Hill

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** suppose I put a file a.txt in package com.xyz and the try access it like following. Will it work?

```
import com.xyz.*;
public class Hello{
  File f = new File("a.txt");
  ...
}
```

it is not working for me. Is there any workaround?

Answer: If the source and the text file are in the jar file, then you access the file by:

```
URL fileURL = getClass().getResource("file.txt");
```

You can then read the file by using a reader (or whatever you choose), e.g.:

```
_istream = new BufferedReader( new InputStreamReader(fileURL.openStream()) );
```

--
j o h n e w e b e r

Or, simpler
getClass().getResourcesAsStream("file.txt"), but you must be sure that file is in the same directory (package) as your class, otherwise you need play with getClassLoader().getResourceAsStream(<path>"/file.txt");
--
Oleg

- **Question:** Difference between loading and instantiating a class?
Well, the subject says it all. What is the difference between loading and instantiating a class in a JVM?

Second **Question:** What would happen if at runtime I update a class file? Will the JVM know to use that instead?

Answer: The difference is that when a class is loaded by a ClassLoader it is read in as a stream of bytes, presumably from a file, but it could just as easily be from over the network, and then processed or "cooked" into a representation that the VM can use to make instances of Objects of that classes type. This last part is the instantiation. You can load a class at runtime with:

```
Class.forName( "MyClass" );
```

and instantiate one with:

```
MyClass mc = Class.forName( "MyClass" ).newInstance();
```

Cool, ehh. You don't have to know the name of a class at compile time.

Answer to second Question**:** What would happen if at runtime I update a class file?

Loaded classes are cached because it's quite costly to do the "cooking" I mentioned above. So it will not be loaded. You may create a separate ClassLoader with new

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

SecureClassLoader but that will cause all classes _it_ loads to be loaded from this new ClassLoader but that's not what you want.

I don't know if you can specify that a class should be loaded from disk again using the normal ClassLoader. You could very easily make your own ClassLoader in which case you would have explicit control over such things. Look at java.lang.ClassLoader and java.lang.Class.
--
Michael B. Allen

- **Question:** Why developers should not write programs that call 'sun' packages?

Answer: Java Software supports into the future only classes in java.* packages, not sun.* packages. In general, API in sun.* is subject to change at any time without notice.

A Java program that directly calls into sun.* packages is not guaranteed to work on all Java-compatible platforms. In fact, such a program is not guaranteed to work even in future versions on the same platform.

For these reasons, there is no documentation available for the sun.* classes. Platform-independence is one of the great advantages of developing in the Java programming language. Furthermore, Sun and our licensees of Java technology are committed to maintaining backward compatibility of the APIs for future versions of the Java platform. (Except for code that relies on serious bugs that we later fix.) This means that once your program is written, the class files will work in future releases.

For more details, see the article Why Developers Should Not Write Programs That Call 'sun' Packages.
http://java.sun.com/products/jdk/faq/faq-sun-packages.html

- **Question:** Can garbage collector remove my singleton? Part 1
A usually singleton

```
public class Single{
  private static Single single;
  private Single {}
  public static Single getInstance(){
    if(single==null){
      single = new Single();
    }
    return single;
  }
}
```

Well, seems good?

But classes are objects too...so do Java 2 v1.3 class garbage collecting? Meaning my singleton could disappear if I don't keep a reference to it (or the class itself) somewhere?

If classes are not garbage collected, that's pretty stupid, I don't want classes taking up memory when I perhaps never will use it again...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: No. Classes can define objects. That is, only the dynamic part of the class defines objects. The static part exists only in one place in memory and can not be duplicated. You can call the getInstance() method from anywhere in your program.

Java requires however that you tell where to find the method, in this case in the Single class.
Therefore, you should use

Single.getInstance()

to get the instance. This is (though it looks much like it) not an execution of a method on an object, but just a method call without object. Single is only used to find out which getInstance() method should be used, and where it is.

You could add a delete() method if you don't need the instance anymore:

```
public class Single{
  private static Single single;
  private Single {}
  public static Single getInstance(){
    if(single==null)
      single = new Single();
    return single;
   }

  public static delete(){
    single = null;
  }
}
```

The garbage collector can now remove the single object after delete() is called if memory is needed.
--
Rijk-Jan van Haaften
Dept of Computer Science, Utrecht University, the Netherlands

Question: Can garbage collector remove my singleton? Part 2
P.S by John:

Doing deeper investigation of this question I found one very good article about this topic. Everybody who are interested can read full article here:

http://developer.java.sun.com/developer/technicalArticles/Programming/singletons/

For the rest of our audience shortly:
A Singleton class can be garbage collected and when
".. a Singleton class is garbage-collected and then reloaded, a new Singleton instance is created. Any class can be garbage-collected when no other object holds reference to the class or its instances. If no object holds a reference to the ' Singleton object, then the Singleton class may disappear, later to be reloaded when the Singleton is again needed. In that case, a new Singleton object will be created.

Any static or instance fields saved for the object will be lost and reinitialized.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

This problem exists in older Java TM Virtual Machines1. JDK 1.2 VMs, in particular, conform to a newer class garbage collection model that forbids any class in a given classloader to be collected until all are unreferenced".

And you "... can also set your VM to have no class garbage collection (-Xnoclassgc on the JRE 1.3, or -noclassgc on the IBM JVM). Keep in mind that if you have a long-running program that frequently reloads classes (perhaps through special class loaders such as the remote class loaders), you have to consider whether that could cause a problematic buildup of garbage classes in the VM."

Also some people asked what is a Singleton and what is relation has it to Java. Shortly a Singleton is one of classical design patterns that are used in software development. More please read in free book here:

http://www.patterndepot.com/put/8/JavaPatterns.htm

Ł **Question:** I study patterns now and would be lost and forget everything very soon!

I am very new to design patterns and just bought the GoF book to learn about it. But as I complete the first couple of chapters right away, I see that they have 23 different patterns and I would be lost and forget everything very soon if I sequentially (page by page) read the entire book!

Do any of you have recommendations on how to proceed with this book so that I 'll remember at least some of them by the time I am done with the book? I can see that many of the classes in java API use composition, facade etc...

But other than that I don't think I will be able to gather anything unless I am able to remember where & when to use particular pattern A, B or C...

Answer: Glad to hear you got the GoF book, it's a great reference manual for patterns. As you've found, however, it's a bit heavy to just "read." What I recommend to people is that they pick a few of the easier, more commonly used patterns: Singleton is a no-brainer that pops up a lot. Adapter tends to get used here and there and isn't that difficult to understand. If you're doing Swing, then definitely go learn the Observer pattern. It'll help to keep you from mixing data and interface code.
Once you've learned three or four and have used them a few times, then as you start new projects; look back to the text to see if there are opportunities in your project where other patterns can be used.

You'll find that over time you'll use more and more of the patterns (some a lot more than others, obviously). I've often found cases where I missed a pattern during design and had "the light go on" after I'd written a bunch of code and realized I was either using a known pattern by accident, or could have used a pattern to my advantage. When possible, I then go back and adjust the design/code to match the pattern.

Keep in mind that the patterns generally don't appear as "absolute." It's expected that you may have variations to satisfy your application's needs.

It's really helpful to others, however, if you make a quick note in your design doc/code about what pattern you were using (which helps them learn patterns too,

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

and helps them understand what you were up to if they know the pattern already).
--
Rico Trooper

- **Question:** When I used java.util.Hashtable, I also used interface Enumeration. But I found that this interface contains defined methods.

  How can an interface contain defined methods? Isn't an interface supposed to hold abstract methods only?

  Answer:
  > How can an interface contain defined methods?
  No.

  > Isn't an interface supposed to hold abstract methods only?
  Yes. (and fields). I'm not sure why you think Enumeration contains defined methods, since it does not (I checked the source to be sure). Keep in mind that an implementing class is also of type Enumeration. So Hashtable's keys method which returns an Enumeration will actually be returning an inner class that implements Enumeration (and therefore defining hasMoreElements and nextElement in a manner that is specific to traversing a Hashtable).

  If this doesn't help, maybe you could clear up what you mean by Enumeration having defined methods.
  --
  Eric

- **Question:** can anyone provide an example on how to use clone () and Cloneable interface?

  Answer:

```
//Class Point uses default clone implementation that returns
//shallow copy of Point (which is sufficient in this case).

class Point implements Cloneable {
    int x, y;
}

class ImmutableRectangle {
  private Point lowerLeft = new Point();
  private Point upperRight = new Point();
  public Point getLowerLeftPoint() {
    try {
      return lowerLeft.clone();
    }
    catch(CloneNotSupportedException e) {
        //shouldn't occur since Point is cloneable
         throw new InternalError();
   }
  }
}
```

we don't want Rectangle points to be changed outside its class. the best way would

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

be to create a copy of Point object and pass it.

i5mast
MVP for Java 2
http://www.brainbench.com

- **Question:** I need to programmatically replace an entry in a zip file.
I could not quite get it using the ZipOutputStream because it simply creates a new file and write only that entry for me. The rest of the original entries are gone.
Does anyone have a solution for this?

Answer:
1) Read the file (myfile.properties) out of the original Zip Archive (original.zip)
2) Make your changes, write the file to the file system
3) Create a New Archive (originalNew.zip)
4) Write your edited file (myfile.properties) to originalNew.zip
5) loop through all the entries in the original Zip archive (original.zip), adding them to the new archive (originalNew.zip) EXCEPT for the file you are replacing (myfile.properties)
6) When you're done, erase the original and rename your new one to original.zip

I believe that this may be the only way to do this, since there doesn't seem to be any random access in the ZIP file.
--
Kevin T. Smith

- **Question:** What is better to use: array or vector?
Just wondering as I am using Vectors to store large amounts of objects from 50 to 4000 and each one has to be "looked at" every time paint is called...

Just wondering if it would be better to use an array, list etc?

Answer 1: Since the Vector method uses an array for storage but has extra steps involved in getting an element, use an array for fastest access.
--
WBB Java Cert mock exams http://www.lanw.com/java/javacert/

Answer 2: arrays are faster, vectors are more dynamic.

This should be evident just looking at the amount of code you need to traverse one versus the other. It might also be beneficial to write a linkedlist class and use that. That way you have a dynamic container which has potential to be faster than a vector (though still not as fast as an array). The problem with arrays is that if you need more space than the current size, you have to hardcode their copying into a bigger array.

Conversely, if you never (or rarely) use the entire array, its a waste of space and memory.

The following are benchmark test results of vector vs. array (ran on a 200-Mhz Pentium w/ 96 Mbytes of memory and Windows95):

Allocating vector elements: 17910 milliseconds
Allocating array elements: 4220 milliseconds

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Accessing Vector elements: 18130 milliseconds
Accessing array elements: 10110 milliseconds

One other reason for vectors being slower that I did not mention above is that vector methods are synchronized, which creates a performance bottleneck.
--
MSW

- **Question:** Would anyone know the performance issues regarding Vector's?
I am actually talking about resource pooling. I have objects that wait in a queue. It is a vector that keeps growing, as the queue gets bigger.

Do Vectors have much performance hit? Is there a better way to implement vectors to get the best out of them? Or am I better of creating a fixed size array?

Answer: If you just want a LIFO or LILO queue, you may be better off with LinkedList than with Vector, as then you'll never have to wait for the contents to be copied.

Vectors perform pretty well, but if you know (even roughly) how big you're going to need it to be, specifying that in the constructor call can help.

How sure are you that this will be a performance bottleneck? Premature optimization is the root of all evil...

The Vector class is thread-safe. By that I mean that there is no way to corrupt the internal representation of the data by accessing the vector from more than one thread. However, it is still possible, very easy in fact, to use a vector in a way that is not thread safe.

Consider this code:

```
for (int i = 0; i < vector.size(); i++) {
   System.out.println(vector.elementAt(i));
}
```

It looks safe, but there's a subtle flaw...

- **Question:** Isn't the Vector class included in Java 1?
I'm writing an applet where I need to use something like ArrayList or Vector. Is there anything in Java1 I can use? When I try using the Vector Explorer gives me the message:

java.lang.NoSuchMEthodError: java/util/Vector:
method add (Ljava/lang/Object;)Z not found

Answer: Before Java2 there was only the method addElement(Object). The method add(Object) was included with the List interface.

I suggest you downloaded the JDK1.1.8 documentation and referred to that (to decide which methods to use).

To avoid such problems in a future please check the 1.1 Vector doc online at:

http://java.sun.com/products/jdk/1.1/docs/api/java.util.Vector.html

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

or, better still, install the 1.1 docs (download from

http://java.sun.com/products/jdk/1.1/docs.html

- **Question:** How do I copy one array to another? Given that I have an byte array defined like this:

```
byte byteSmall = new byte[23];
```

and another larger byte array defined like this:

```
byte byteBig = new byte[30];
```

How do I copy byteSmall into byteBig starting at index 7 without a for loop like this:

```
for(int i = 0; i < 23; i++){
  byteBig[i + 7] = byteSmall;
}
```

Answer: See System.arraycopy:

"Copies an array from the specified source array, beginning at the specified position, to the specified position of the destination array. A subsequence of array components are copied from the source array referenced by src to the destination array referenced by dst. The number of components copied is equal to the length argument. The components at positions srcOffset through srcOffset+length-1 in the source array are copied into positions dstOffset through dstOffset+length-1, respectively, of the destination array.

If the src and dst arguments refer to the same array object, then the copying is performed as if the components at positions srcOffset through srcOffset+length-1 were first copied to a temporary array with length components and then the contents of the temporary array were copied into positions dstOffset through dstOffset+length-1 of the argument array."

- **Question:** More about Robot! I met with a problem in using class Robot.mousePress...

The compiling process is successful. But when I run it, I receive "IllegalArgumentException:

Invalid combination of button flags". I don't quit understand this information. Part of my code is as following:

```
Robot rMouse=new Robot();
int button=1;
rMouse.mousePress(button);
rMouse.mouseRelease(button);
```

I am really confused. Will you please give me some advice? Thank you in advance!

Answer: You are not using a valid value for the argument to the mousePress() and

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

mouseRelease() methods. If you check the API documentation, you'll find the valid values are a combination of one or more of the following constants:

InputEvent.BUTTON1_MASK
InputEvent.BUTTON2_MASK
InputEvent.BUTTON3_MASK

plus others which represent the Ctrl, Alt, and Shift keys. To press the left mouse button, you want to use:

```
rMouse.mousePress(InputEvent.BUTTON1_MASK);
```

--
Lee Weiner

- **Question:** In what situation an exception has to be caught otherwise the compiler will complain?

E.g. IOException does NOT have to be explicitly caught, however, SQLException has to be caught otherwise VisUalAge will not compile the program.

Answer: The only unchecked exceptions in Java are RuntimeException and its subclasses. This includes such familiar classes as NullPointerException, ClassCastException, and IndexOutOfBoundsException.

IOException is not one of these, and *does* have to be explicitly caught or thrown
--
jeff_robertson

- **Question:** Is it possible to use switch with range of values?

Ex:

```
switch (iBegins){
  case 2 to 10:
}
```

Answer:
not exactly like this but:

```
switch (iBegins){
  case 2:
  case 3:
  case 4:
  case 5:
  case 6:
  case 7:
  case 8:
  case 9:
  case 10:
    /* do stuff */
    break;
  case 11:
    /* other stuff */

}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Jan Schaumann http://www.netmeister.org

- **Question:** Is there a general reason why null pointer exceptions occur?

Answer: Of course there is. A NullPointerException happens when you have a reference, its set to null (or is a class or instance variable and has never been initialized), and you try to perform some operation on the object it points to. Since it really doesn't point to an object, you can't do that.
--
Chris Smith

- **Question:** I am aware that the exceptions in the catch statements must be of the same type or a super type of the exception thrown in the try block.

My question is this:
Is there any significance in the ordering of the catch statements after a try block?

Answer: It is required to design you catch statement sequences with most derived exception class type first and progressing to the highest superclass. Else, the code will not compile.

For a obvious reason, if the a catch block for a given class precedes a catch block for a type that is derived from the first, the second catch block will never be executed.
--
D. Michael Nelson

- **Question:** I wrote a program that uses a few RS232 ports. The operators are unskilled and often start multiple instances of the program. Will someone please be so kind and tell me how I can prevent them doing it?

Answer 1: The first instance might write a file. Subsequent instances could check for the existence of that file, or else check its contents.

Another method could involve creating a server socket on a specific port. Subsequent efforts to create a socket on that same port would throw an exception.

Answer 2: Actually a better way is to (on launch):

1) Check if the file exists. If not, create it, open it and run. Leave it open until you quit, upon which time you close it.

2) If the file _does_ exist, try to delete it. If the delete fails, then someone else has it open, which means another copy of your app is running. Inform the user and quit.

3) If you succeeded in deleting it, then you are the first. Now create, open and run.

Doing the above prevents the problem of having the semaphore file left around when the system crashes. I implemented it recently in one of our apps, and it works like a charm.
--
Burt Johnson
MindStorm Productions, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

http://www.mindstorm-inc.com

- **Question:** Can you call a class static abstract method from an abstract class or does it need to be extended and then called from its concrete class?

  I've been told that abstract classes do not actually have any code in them cause they are like a placeholder, so I guess you wouldn't bother calling a static method in an abstract class cause it wouldn't have any code to begin with....?

  Answer: You have been told wrong. Abstract classes can and do have code in them.

  See, for example, java.awt.Component, an abstract class with a lot of code and no abstract methods at all. If a class has any abstract method member, directly declared or inherited, it is required to be declared abstract. If not, it is the programmer's decision and should be based on whether it would make sense to have an instance of that class.

  Perhaps whoever told you was confusing abstract classes with interfaces, which don't contain implementation, only abstract method and constant declarations.

  You cannot declare a method to be both static and abstract. Abstract requires the method to be overridden before you can have a concrete class, static prevents overriding. You can have a static method in an abstract class - such a method could be called without creating an instance of the class, the only thing that is prohibited for abstract classes.

  And when a subclass of an abstract method has been instantiated, all the methods from the original class will keep the same code in the instance. Most of the time an abstract class will have abstract methods.

  However, there are several examples of abstract classes that don't have any abstract methods at all. Some examples are Component and FontMetrics from the AWT. It doesn't make sense to have just a Component that's not a specific type of component. It doesn't make sense to have a FontMetrics that doesn't measure any specific kind of Font.

  Also being abstract never prevents overriding, it just requires overriding in order to derive a non-abstract subclass. And if a class is a subclass of an abstract class, it only MUST override those methods declared abstract. The other methods do not require overriding.

- **Question:** Why can not I compile this simple example?

```
public class Test{
   int a;
   final static int fin;
   final int finint;
}
```

  Answer: You can not compile it because you have to initialize final variables. It must be done explicitly.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I write java about 2 years, but I still confuse one thing that is why should we use interface???

  If I need to implement an interface and just use its every methods name. Why shouldn't just write every methods statements in a class, not in interface?

  I only can think about that if I extend a class, than can implement another or the others interface.

  As you can saw, I really confuse about this. And I do see many books for the reasons, but I can't get the answer, please tell me!

  Answer: "Interface" is the Java way to do multiple inheritances, or a better way to think of it is as a way to design plug-ins.

  For example, let's say we have an application that monitors a network of computers. Our monitors might check for web pages, or they may check for other ports, or they may have hooks for hardware checks.

  The interface to our main control panel is always the same: We need some means to poll the monitor object for an answer. This is the "NetworkMonitor" interface and all network monitors will share this interface, but they may have a class hierarchy that is very different, for example, port-monitors may all fork a thread that periodically checks whereas our control panel interface just asks for the most recent answer; hardware monitors may ask for their data in real-time or over RPC and thus have no need of inheriting from Thread.

  Because they share the same Interface definition, the control panel application does not need to know if they are polling monitors or real-time monitors because, from the control panel's point of view, it does not matter
  --
  P.S. by John
  Also interfaces make our life (I mean programmers) much easier.
  Imagine a big project (a lot of programmers, many packages, thousands of files): it is impossible all the time to be aware about particular implementation of some method in specific class in another package!

  Much easier to define interfaces between packages as some kind of bridges or gates into another package and their functionality, with hidden (for us) implementation. We just know interface and method names. It is enough to use those methods.

  How it is implemented there does no matter... It is working!

- **Question:** I find such function, but in Class class. Is it possible to use it (or something like this) in another class? Exactly I would like call:

  super.super.method1();

  Answer: No, you can only access methods of your direct ancestor. The only way to do this is by using reflection:

  import java.lang.reflect.*;

  Class superduper = this.getClass().

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
getSuperClass().
getSuperClass();
Method ssMethod = superduper.getMethod("method1",null);
ssMethod.invoke(this,null);
```

(I didn't check the exact syntax and parameters of the above code.)

However the necessity to skip a hierarchy shows that there is something wrong with the design of your application. You should fix this instead of using dirty tricks like the above.

Carl Rosenberger
db4o - database for objects - http://www.db4o.com

- **Question:** I am facing a problem that might sound common to you all. I have to write a switch statement with 40 case statements. I am wondering if there can be some way to reduce this. If anybody can provide some tips on how to design this smartly. ;)

  Answer: The alternatives are:

  - an array you index into to get values or delegates.
  - an array you binary search.
  --
  Roedy Green
  http://www.mindprod.com or http://209.153.246.39

- **Question:** Does anyone know what the character limit for a class name would be? I have my class name in a variable ft and I am doing a Class.forName(ft);

  My class name with qualifiers is 44 characters long, I can only imagine that there must be a limit ...

  Answer: From the JVM Spec 2 - 4.2

  Class and interface names...fully qualified name...as
  CONSTANT_Utf8_info.

  That structure, CONSTANT_Utf8_info, places a limit of 64k on the length. But for a class this is the fully qualified name, so the maximum length of the name could be less.
  --
  jschell

- **Question:** I read it on:

  http://java.sun.com/docs/books/tutorial/java/javaOO/override.html:

  "Also, a subclass cannot override methods that are declared static in the superclass. In other words, a subclass cannot override a class method. A subclass can HIDE a static method in the superclass by declaring a static method in the subclass with the same signature as the static method in the superclass. "

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

My **Question:** It looks like play with words: override, hide.. Why is it written HIDE?

If I can write a static method in subclass with the same signature it means that I override that method in superclass, I really changed that method here in subclass. Am I wrong?

Answer: Let's start from the very beginning - definition for overriding. It includes three main points that overridden method must:

* have the same name
* have the same signature
* have the same data type.

In your case the static method in subclass can have another data type.
Example:

```
************ Alex.java *************
package alex;
public class Alex {
  static String al(String name){
    return name;
  }
}
********************* EOF ***********

************ John.java *************
package alex;
public class John extends Alex {
  static int al(int name){
    return name;
  }
}
********************* EOF ***********
```

It compiles! You see that in Alex class the method "al" returns String and in John class - int.

It is hiding, not overriding.

Advice: try to avoid such situation when you hide methods in super class. Use different names instead and you will save a lot of time.

- **Question:** Why C++ is not platform independent?

  Answer: C++ compiles to binary code (.obj, .exe, .dll, a.out etc.). Binary code (Machine code, 0's and 1's) are machine dependent.

  Java compiles to byte code which is independent of any machine. It need to be interpreted to binary code by JVM, and executed by the machine

- **Question:** I know that void method does not return any value. But I still write the code like this:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
void nothing() {};
void nothing2() {return;};
```

Why can we still use "return" in the body of method?

Answer: To be able to exit method when it is necessary.
You can exit the method when you reach the end of method and do not need to use "return". If due to some condition program must exit the method then you use "return".

- **Question:** I can't find the API documentation on any classes in the sun.* packages. Where is it?

Answer: The short answer is that SUN provides documentation only for the public classes in java.*. SUN does not provide documentation for sun.* because those are the Sun-specific implementation, and specifically not part of the Java technology API standard, and are therefore subject to change without notice.

In general, SUN doesn't provide javadoc documentation for sun.* classes in order to discourage developers from writing programs that use them. For further explanation, see the next question.

However, if you must have it, the documentation for sun.* is available separately, here: http://java.sun.com/communitysource/index.html
For example, the doc comments for sun.net are in the source files located at:

/src/share/sun/sun/net/*.java

This source code release does not include javadoc-generated documentation. You would have to generate those docs yourself using javadoc.

source: http://java.sun.com/products/jdk/faq.html#A12

- **Question:** Where can I get source code for Java 2 platform?

Answer: In most cases you do need the code itself. Often it is enough all those tools and resources that are included into standard Java SDK.

Anyway you can always find it here:
http://www.sun.com/software/communitysource/java2

- **Question:** Does the java class that call the JNI need to be in the same directory of the DLL?

I have the java class in a package and in a jar file. The DLL it uses is outside of the jar file. The JNI call does not work. I am confused with the path.

Answer: As long as the library is in your "path" you should be able to load it with java.lang.System.loadLibrary.

On Linux it should be in your LD_LIBRARY_PATH and on Windows in your PATH. With java.lang.System.load you can load the library from a predefined location.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

--
Flaps

- **Question:** ActionListener is an interface and we cannot instantiate an interface. Then how is it possible to do something like this (see below)...?

```
textField[0].addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent e){
      handleSearch();
   }
});
```

Answer: That notation is essentially shorthand.
The compiler fills in the equivalent of
XXXX extends Object implements ActionListener {
}
--
William Brodgen

- **Question:** By default an application has no security manager and Java runtime environment does not create automatically a security manager for my program. How then applet where I am not creating any security manager already prevented from many operations?

Answer: It is true - Java by default let you do whatever you want and then it is your responsibility to restrict something. In case with applets a little bit different story - applet viewers and browser have THEIR OWN security manager, not from JRE.

That's why even if you did not define any security manager in the code of your applet, during the start up time browser/viewers will use their own security manager. This manager is built in into their application (browser, viewer), not your Java applet.
--
AP (J.A.)

- **Question:** Does java support ASM?

Answer: You can't directly embed ASM code in your java source. However, you could just use the Java Native Interface to call a C (or whatever else you like) library that embeds the ASM code. This will make you platform-dependent!
--
Sebastian

- **Question:** I am a Java beginner. I just want to ask what's the difference between

```
int [] number;

  and

int number[];
```

Answer: The "postfix []" syntax is to make the language more comfortable with C programmers. In Java, the "postfix []" binds to the variable name, while the "prefix

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

[]" binds to the type name.

In C, there is only the "postfix []". But C also includes pointers, which Java doesn't. The pointers are identified by having a * between the type name and the variable name.

While there is no danger of ambiguity in this declaration:

int a[], b;

there IS danger of ambiguity in this declaration:

int *a, b;

In the latter case, the C rules are that a is declared a pointer, while b is not.

Java offers the following different ways of declaring arrays:

int a[], b; <--> int[] a, b;

These are not equivalent, because b is an array only on the right side. Whereas with C-style pointers these would become:

int *a, b; <--> int* a, b;

which breaks the concept of "white space doesn't matter", so it will not be as easy to scan and parse.

So the designers of C decided to have only 1 form of defining variables with pointer types or array types, and this was that the * or the [] binds to the variable, not to the type.

Java carried over the variable binding, but also introduced the type binding, because there were no more pointers, so the ambiguity was removed.
--
Joona Palaste

And think also about marketability!
Making Java's syntax highly similar to C and C++'s during its early years:

* Made the language seem more familiar to C/C++ programmers.
* Decreased the learning curve for C/C++ programmers.
* Gained credibility through the frequent inference that it was a next-generation descendant of C++.
* Increased its pool of programmers by attracting C/C++ programmers through the effects above.

This isn't the only instance of anachronistic syntax; consider the optional semicolon at the end of a class declaration.
--
Andrew R.

Ł **Question:** Besides speed, what does the IBM JDK have that sun's does not? Are there any compatibility problems?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: If there are problems, then they represent bugs in one JDK or the other. To the developer, they should present identical APIs and behavior.
--
Jim Sculley

- **Question:** I've seen a few examples were for loops are used in different way

```
for(;;) {
}
```

does this do the same as a

```
while(true)
```

is this a speed issue?

Answer: The two are identical. It's simply a matter of style and personal taste.
--
Kevin Riff

- **Question:** When do I need to use overloading in a program? And why?

Answer: Overloading is best used when you want to have a function that has the same name but can accept different arguments. The only restriction is that it must return the same type.

e.g.

```
public String aName(String str) {
    String retStr;
    retStr = "Hello" + str;
    return retStr;
}
// Now the same function but overloaded
public String aName(String str, String str2){
    String retStr;
    retStr = "Hello " + str + str2;
    return retStr;
}
```

Both functions return the same type (a String) but have different signatures.

You will find this used a lot with constructors. See any recommended text on Java for a much better explanation.
--
Anthony Miller

- **Question:** Why people claim that java is a secure language...

Answer: The java programming language itself is just as secure or insecure as any other programming language. A Java Applet (which are used on the web) is a different matter. They are quite secure.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

An applet does not have access to system resources outside of the directory that it is created in. There are ways that you can give an applet access to this information, but they are pretty explicit and you will most likely know that the applet is trying to do this.

I am not really familiar with the subject but did a little reading on the sun website. Check it out for more info:

http://developer.java.sun.com/developer/technicalArticles/Security/

You aren't supposed to be able to break the rules easily at run time. The elimination of pointer arithmetic is supposed to improve type safety.

The Code Red worm is a good example of the weakness of the 'old-style' languages. This worm uses a 'buffer overflow' attack to 'trick' the IIS program in to doing it's bidding.

The Java system is designed to make it harder to make the programming mistake that Code Red exploits. It is important that you understand that Code Red would not have been possible without careful attention to a fast buck by Mr. William Gates III and his cohorts.
--
Dr Hackenbush

- **Question:** This is a basic question... While looking at java API s I noticed that there are classes that start with java and javax (like java.lang.util). What is the basis for this?

  Answer: The size of the core part of the Java platform has been growing steadily since the release of version 1.0. The first Java platform had 8 core packages, in version 1.1 there were 22 packages, and in version 1.2 there are over 50! The extensions framework provides a standard means to add functionality to the Java platform for use when needed, without having to increase the size of the core API. Standard extensions have names in the javax.* namespace.

  However, just because a package name begins with javax does not guarantee that that package is an extension rather than a part of the core platform. The Swing packages have names beginning with javax indicating that they were non-core extensions prior to version 1.2 of the platform. They are part of the core 1.2 platform even though they are in the javax.* namespace.

  For more details please see:
  http://java.sun.com/products/jdk/1.2/docs/guide/extensions/ext_faq.html

- **Question:** Does anyone know if there is a way to prevent System.exit() being called?

  Answer: Look into java.lang.SecurityManager. It has a method called checkExit() with which you can prevent System.exit().

  This method throws a SecurityException if the calling thread is not allowed to cause the Java Virtual Machine to halt with the specified status code.
  --

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

David Zimmerman

- **Question:** I want forcibly terminate current running JVM and do not want initiate its shutdown sequence. Just STOP it!

  Answer: Use method halt() (Runtime) since: it forcibly terminates the currently running Java virtual machine. This method never returns normally.

  This method should be used with extreme caution. Unlike the exit method, this method does not cause shutdown hooks to be started and does not run uninvoked finalizers if finalization-on-exit has been enabled. If the shutdown sequence has already been initiated then this method does not wait for any running shutdown hooks or finalizers to finish their work.

- **Question:** What is difference between virtual-machine shutdown hook and using the method exit()?

  Answer: Sophisticated applications often require the ability to perform specific actions when the virtual machine shuts down, whether due to voluntary exit or involuntary termination. Typical actions include cleanly closing down network connections, saving session state, and deleting temporary files.

  Every major operating system provides the means for native applications to do this, though of course the details vary from platform to platform.

  Shutdown hook is just one of three steps that executed when you run exit() method.

  Terminates the currently running Java virtual machine by initiating its shutdown sequence. This method never returns normally. The argument serves as a status code; by convention, a nonzero status code indicates abnormal termination.

  The virtual machine's shutdown sequence consists of three phases:

  - In the first phase all registered shutdown hooks, if any, are started in some unspecified order and allowed to run concurrently until they finish.

  - In the second phase all uninvoked finalizers are run if finalization-on-exit has been enabled.

  - And finally when this is done the virtual machine halts.

- **Question:** What should I take into consideration when I add my Shutdown hook?

  Answer: If you look at the signature of the method:

  public void addShutdownHook(Thread hook)

  you see that this method uses a thread as an argument. A shutdown hook is simply an initialized but unstarted thread. When the virtual machine begins its shutdown sequence it will start all registered shutdown hooks in some unspecified order and let them run concurrently. When all the hooks have finished it will then run all uninvoked finalizers if finalization-on-exit has been enabled. Finally, the virtual

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

machine will halt. Note that daemon threads will continue to run during the shutdown sequence, as will non-daemon threads if shutdown was initiated by invoking the exit method.

Once the shutdown sequence has begun it can be stopped only by invoking the halt method, which forcibly terminates the virtual machine.

Once the shutdown sequence has begun it is impossible to register a new shutdown hook or de-register a previously-registered hook. Attempting either of these operations will cause an IllegalStateException to be thrown.

Shutdown hooks run at a delicate time in the life cycle of a virtual machine and should therefore be coded defensively. They should, in particular, be written to be thread-safe and to avoid deadlocks insofar as possible. They should also not rely blindly upon services that may have registered their own shutdown hooks and therefore may themselves in the process of shutting down.

Shutdown hooks should also finish their work quickly. When a program invokes exit the expectation is that the virtual machine will promptly shut down and exit. When the virtual machine is terminated due to user logoff or system shutdown the underlying operating system may only allow a fixed amount of time in which to shut down and exit. It is therefore inadvisable to attempt any user interaction or to perform a long-running computation in a shutdown hook.

In rare circumstances the virtual machine may abort, that is, stop running without shutting down cleanly. This occurs when the virtual machine is terminated externally, for example with the SIGKILL signal on UNIX or the TerminateProcess call on Microsoft Windows. The virtual machine may also abort if a native method goes awry by, for example, corrupting internal data structures or attempting to access nonexistent memory. If the virtual machine aborts then no guarantee can be made about whether or not any shutdown hooks will be run.

Based on Java API description

Ł

8. Java Bugs

Question: The clear-Method in the java.util.Calendar-Class doesn't behave like I suggest. I want to clear the hour-field in a Date. If I use Calendar.clear(Calendar.HOUR) the hour is not reset. If I use

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Calendar.clear(Calendar.HOR_OF_DATE) the hour is not reset also. Only if I invoke both methods the hour is reset. (java version "1.2.2")

Answer: It is known Java bug.

Call cal.clear(cal.HOUR) as well as cal.clear(cal.HOUR_OF_DAY).

or

Call Calendar.set(Calendar.HOUR_OF_DAY,0) instead of Calendar.clear

(Calendar.HOUR_OF_DAY).

This bug is not fixed yet...

9. Java Hardware

Question: Can anyone please direct me to some literature pertaining to the Java chip.

Is the project still being pursued by Sun? I remember hearing about it in 2000 but haven't heard of any companies adopting the chips. This makes me worry about the chip's cost and availability.

Answer: You can find some companies producing Suns java chip and different ones on the link section of http://www.jopdesign.com

Sun has had at least two "Java chip" projects that I know of.

PicoJava was the earlier one -- I thought it died quietly, but I see that Fujitsu lists a "32-bit PicoJava-II microcontroller", so maybe not. The more recent one is MAJC, but it isn't aimed at small or low-power devices.

New ones keep coming out of the woodwork, e.g. this announcement yesterday:

http://dailynews.yahoo.com/h/nm/20010604/tc/tech_omron_chip_dc_1.html

If you want to experiment with one that's available now and does real-time Java, see:
http://www.jstamp.com/

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

A fair number of us have also used the TINI board
http://www.ibutton.com/TINI/

though it is aimed at industrial controllers, not PDAs. (It's a conventional CPU running a JVM, not a "Java chip", but that may be a silly distinction).

Imsys makes a Cjip: http://www.imsys.se/

http://www.ptsc.com have had their PSC1000A 32bit 100MHz $10 'Java' chip available for some time. I've been using it (but not with Java) for over a year. It rocks.

They also make a complete module (using a PSC1000A) called Ignite1.
--
Thomas Maslen

10. Java Tools and Software

Question: I often do mistakes when write my JavaDoc comments. Do you know a tool to avoid it?

Answer: Yes, I know, at least one: iDoc - the javadoc documentation QA tool. On iDoc's web site:

> iDoc warns about missing and obsolete @param and @return tags in method comments. This enables you to keep documentation and code in synch.
>
> iDoc finds typos in tag names (e.g. if you typed @versuon instead of @version). This guards against missing documentation entries due to typos.
>
> iDoc finds comments that do not have a descriptive "title". In addition this also points out classes, interfaces, methods and instance variables that do not have a javadoc code comment at all.
>
> iDoc is a "smart grep" utility that can search for specific comment entries in your source code (e.g. list all methods that contain a "@open ..." tag in their comment). This enables you to introduce your own tags, such as "@open", "@rework", etc.
>
> iDoc lists all occurrences of tags that are not part of the standard javadoc tag set (@author, @version, @param, etc.).

It is free!--
John

Question: What is a new feature in Java 1.5 that names "autoboxing"?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: As you know, the Java programming language has a "split type system": some types are primitives and others are object references. You can't put primitives into collections, so you end up converting back and forth between primitives (such as int) and "wrapper types" (such as Integer) when you need to store them in collections. Anyone who has done this can attest that it isn't pretty.

From "A Conversation with Joshua Bloch" article at:
http://java.sun.com/features/2003/05/bloch_qa.html

For example, take a look at this program, which generates a frequency table of the words found on the command line. It uses a Map whose keys are the words and whose values are the number of times that each word occurs on the line:

```
public class Freq {
   private static final Integer ONE = new Integer(1);
   public static void main(String args[]) {
      // Maps word (String) to frequency (Integer)
      Map m = new TreeMap();

      for (int i=0; i<args.length; i++) {
         Integer freq = (Integer) m.get(args[i]);
         m.put(args[i], (freq==null ? ONE :
              new Integer(freq.intValue() + 1)));
      }
      System.out.println(m);
   }
}
```

Notice how messy the inner-loop code that increments the count looks? Now take a look at the same program rewritten with autoboxing, generics, and an enhanced for loop:

```
public class Freq {
   public static void main(String args[]) {
     Map<String, Integer> m = new TreeMap<String, Integer>();
     for (String word : args) {
       Integer freq = m.get(word);
       m.put(word, (freq == null ? 1 : freq + 1));
     }
     System.out.println(m);
   }
}
```

Sweet, huh? One thing worth noting: this program assumes that when you auto-unbox null, you get zero. It's still an open issue whether this will be the case. The alternative is to throw NullPointerException. Both alternatives have their advantages. Unboxing null to zero beautifies applications like the one above, but it can also sweep real errors under the rug. If anyone has any strong opinions, or better yet, convincing arguments on this issue, please pass them along to the JSR-201 expert group.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

From "A Conversation with Joshua Bloch" article at:
http://java.sun.com/features/2003/05/bloch_qa.html

In the article you will find more descriptions of coming changes in Java 1.5

Question: How operator "for" will be enhanced in Java 1.5?

Answer: I give the answer of Joshua Bloch from article at:
http://java.sun.com/features/2003/05/bloch_qa.html

In the article you will find more descriptions of coming changes in Java 1.5.

Iterating over a collection is uglier than it needs to be. Most of the time when you iterate over a collection, you don't use the iterator for anything except getting elements. The "enhanced for" statement makes the compiler take care of the iterator for you. For example, here's a method that traverses a collection of timer tasks using an iterator:

```
void cancelAll(Collection c) {
   for (Iterator i = c.iterator(); i.hasNext(); ) {
       TimerTask tt = (TimerTask) i.next();
       tt.cancel();
   }
}
```

Now here's the same method with an "enhanced for" statement:

```
void cancelAll(Collection c) {
   for (Object o : c)
       ((TimerTask)o).cancel();
}
```

When you read the statement out loud, the colon (:) is pronounced "in." It might have been more natural to use two new keywords, foreach and in, but new keywords are destabilizing. They break existing sources that used the new keywords as identifiers. We went out of our way to preserve compatibility when we specified the new language extensions.

Question: How does new Java (1.5) provides compile-time type safety for collections?

Answer: This is done with the new feature that names "generics"...

Answer was done the article at:
http://java.sun.com/features/2003/05/bloch_qa.html by Joshua Bloch.

In the article you will find more descriptions of coming changes in Java 1.5

Here's how you do it today:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
/**
 * Remove the four-letter words from the specified
 * collection, which must contain only strings.
 */
static void expurgate(Collection c) {
    for (Iterator i = c.iterator(); i.hasNext(); ) {
        String s = (String) i.next();
        if(s.length() == 4)
            i.remove();
    }
}
```

The casts aren't very pretty, and more importantly they can fail at run time. Suppose the user accidentally passes in a collection that contains string buffers rather than strings. The comment says that the client is required to pass in a collection of strings, but the compiler can't enforce the comment.

Here's how the same method looks with generics:

```
/**
 * Remove the four-letter words from the specified collection of strings.
 */
static void expurgate(Collection<String> c) {
    for (Iterator<String> i = c.iterator(); i.hasNext(); )
        if (i.next().length() == 4)
            i.remove();
}
```

Now it's clear from the method signature that the input collection must contain only strings. If the client tries to pass in a collection of string buffers, the program won't compile. And notice that the method doesn't contain any casts. It's one line shorter and, once you get used to reading generics, it's clearer, too.

Question: What is Java Web Start Technology?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: Java Web Start allows you to launch Java-technology-based applications directly from the Web. An application can be launched in three different ways:

- from a Web browser by clicking on a link.

- from Java Web Start's built-in Application Manager, which tracks recently used applications and provides quick access to your favourite applications.

- from desktop icons or the Start Menu (Microsoft Windows only).

Regardless of which way is used, Java Web Start will connect back to the Web server each time an application is launched to check whether an updated version of the application is available.

Java Web Start technology can automatically create shortcuts on the Microsoft Windows desktop and in the Start Menu for Web deployed Java-technology-based applications. By default, Java Web Start asks the second time an application is launched, if a shortcut should be created. This can be changed using the Preference panel.

Shortcuts can also be added and removed by using the Application Manager, using the Application/Create Shortcut, and using Application/Remove Shortcut menu item.

Check SUN's demos here: http://java.sun.com/products/javawebstart/demos.html

From SUN's description

Question: Is Web Start a general installer?

Answer: Web Start is different from classic, big blue gradient, pre-Internet, install-and-forget-it, single-shot installers. Web Start is designed for Java apps and, therefore, isn't of any help if you want to install Windows .exe apps that require a bunch of registry settings, for example.

Question: Can I use Web Start to deploy apps to mobile devices (J2ME)?

Answer: Web Start is designed for rich desktop apps (aka resource hogs) requiring Megas of disk space and very likely doesn't fit on your Java phone unless you attach a Giga hard disk somehow.

If you want to deploy applications to mobile devices consider using MIDlets.

http://lopica.sourceforge.net/faq.html

Question: I found an option in Java hotspot VM: -UseISM. What is USM and where do I need it?

Answer: Intimate Shared Memory (ISM) is a Solaris facility that can help by:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

- Enabling larger pages - 4MB instead of the default 8KB
- Locking pages in memory - no paging to disk
- Allowing shared virtual to physical address data structures

The 1.3.1 JVM exploits the first two. By reducing the overhead of virtual to physical address translation, a significant performance boost can often be obtained (typically 5-10%) . However, not all applications improve with ISM (it is possible to get worse), so it is important to monitor performance and resource use.

On Solaris 9 and JVM 1.4.1 or later, ISM is not usually needed; a similar facility called MPSS is enabled by default. This works like ISM but without locking the pages in memory or needing the other modifications below. ISM can still be used if desired.

You need to have SUNs machine with Sparc V9 (ultrasparc I or greater) CPU.

To use ISM, add the following to the /etc/system file:

set shmsys:shminfo_shmmax=0xffffffff

set shmsys:shminfo_shmseg=32

If your system is 2.6, you may also need to add

set enable_grp_ism=1

After those modifications, reboot and run the VM with the flag

-XX:+UseISM

Using ISM can cause certain problems. How to avoid them read here:

http://java.sun.com/docs/hotspot/ism.html

Question: Can I use Java Web Start for distributing Web Start applications from CDs, without putting them to web server?

Part 1.

Answer: Yes, you can do.

Vamp (aka Venus Application Publisher) offers a couple of choices such as installing your app directly into Web Start's cache or using single, self-contained jars that include a built-in, ultra light-weight web server that serves up your app's jars from the installer's jar itself.

Clio - single-jar installer (app served up through built-in web server)

http://lopica.sourceforge.net/clio.html

Jess - single-jar installer (app installed directly into Web Start cache)

http://lopica.sourceforge.net/jess.html

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Celia - command line installer to install web archives (.war) or client archives (.car) directly into Web Start cache

http://lopica.sourceforge.net/cache.html

Question: Can I use Java Web Start for distributing Web Start applications from CDs, without putting them to web server?

Part 2.

Pam - package manager that installs web archives (.war) or client archives (.car) directly into Web Start cache

http://lopica.sourceforge.net/pam.html

Nullsoft Installer plus Celia (Windows Only) - single Windows (.exe) that installs app directly into Web Start cache

http://lopica.sourceforge.net/winoffline.html

Client archives (.car) are standard jars that include all required jars, icons as well as a all required JNLP descriptors including extensions that are needed by an app in a single jar similar to RPM packages.

http://lopica.sourceforge.net/faq.html

Question: Can I use Web Start for command line/batch apps? Part 1.

Answer: Support for command line/batch apps is currently limited. You can launch apps from a command prompt/shell. Example:

javaws http://java.sun.com/apps/draw.jnlp

However, you cannot pass in arguments for your app on the command line as javaws only takes a single URL as an argument and ignores the rest.

If you want to pass in arguments to your app, you have to add them to the start-up file (aka JNLP descriptor) using <argument> or <property> elements. Nothing is wrong with "hard-coding" your arguments in the start-up file (aka JNLP descriptor) as long as your arguments don't change.

If your arguments are not known in advance or change from time to time, there are a couple of workarounds to pass your arguments along to your app.

A simple workaround is to create a servlet that takes in your arguments as parameters in the URL (e.g. http://localhost/apps/notepad.jsp?dir=c:/carrie/chap1/shower.txt) and adds them to the startup file (aka JNLP descriptor) that it sends along to Web Start. This workaround requires a web server.

(One trick is to make sure not to include the href attribute in the JNLP file that your servlet sends back to Web Start. This will tell Web Start to disable the update check

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

on JNLP files, and Web Start will not treat each new JNLP file as an application update - only updated jar files will.)

Question: Can I use Web Start for command line/batch apps? Part 2.

A more esotereric workaround that doesn't require a web server is creating your own JNLP client that allows you to pass along your arguments to your apps. That sounds harder than it is. Building a stripped down JNLP client (e.g. no installer, no applets) using one of the two open-source client as a start takes probably just a couple of days and should be sufficient for in house usage.

A more sophisticated workaround that doesn't require a web server and that doesn't require you to replace Web Start is wrapping your own executable around javaws that takes your passed in arguments plus JNLP URL and looks up the original in the cache and patches it or adds a new one to the cache that it passes on to javaws to start up the app using your arguments.

The next step would be to create another executable that wraps your wrapper to make a command-line call look like as nothing had happened. Example:

javawsex http://jakarta.apache.org/ant.jnlp jar javadoc crossref

javawsex http://java.sun.com/apps/notepad.jnlp c:/carrie/chap1/shower.txt

becomes ant jar javadoc crossref
notepad c:/carrie/chap1/shower.txt

http://lopica.sourceforge.net/faq.html

Question: How does Web Start differ from Applets?

Answer: Web Start != Applet 2.0/NT. Web Start is not a replacement for applets unless you abused applets for mega-sized pop-up windows floating outside of web pages. Applets run inside the web page on the browser's one and only Java runtime. In contrast Web Start apps run on its own Java runtime outside the browser.

Applets - dynamic content inside a web page in a browser Web Start Apps - full-blown, stand-alone desktop apps that run without a browser

http://lopica.sourceforge.net/faq.html

Question: Can Java Web Start resume download?

Answer: Web Start doesn't support "resume download" on individual jars. However, Web Start will resume downloading the next jar that was not completed. Thus, the best work around for mega downloads over dial-up connections, for example, is to break up your app into several smaller jars instead of using a few biggies.

Question: Can Web Start do background updates and notify the user to restart her app to get the newest version?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: You cannot check for updates in running app using the Web Start API. Web Start only checks for updates on startup.

Question: The Java compiler and other Java tools can only process files which contain Latin-1 and/or Unicode-encoded (\udddd notation) characters. native2ascii converts files which contain other character encodings into files containing Latin-1 and/or Unicode-encoded characters.

Where can I find a tool to convert a file with native-encoded characters (characters which are non-Latin 1 and non-Unicode) to one with Unicode-encoded characters?

Answer: In your JSDK's bin directory! It has the name - native2ascii.

You should be aware that this tool does not support all worlds' languages, just those that are listed on this page (1.4.2):

http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html

Good to use: I found NetBeans IDE and Platform - an open source project.
"NetBeans has positioned itself as a leader in software development market. The products make use of standards (swing), the portability of Java, mature code, and a supportive community. NetBeans software is for free commercial and non commercial use and being used for software development by today's top companies and individuals throughout the world. The software you can find on this site consists of two major pieces:

NetBeans IDE - A world-class, professional IDE (Integrated Development Environment) - The NetBeans IDE is the Platform (see below) plus modules that include things such as an editor, tools for working with source code (Java, C++ and others), version control, and a lot more.
Advanced syntax highlighting, error checking code editor
Support for the Java, C, C++, XML and HTML languages
Pluggable support for compilers, debuggers and execution services
Support for JSP, XML, RMI, CORBA, JINI, JDBC and Servlet technologies
Support for Ant, CVS and other version control systems
Visual design tools
Wizards and code generation and management tools
Cross-platform

NetBeans Platform - Building a desktop application using the NetBeans Platform can save years of developer time. The Platform provides the services common to almost all large desktop applications - window, menu, settings management and storage, file access and more. Developers get the freedom to concentrate on their business logic - writing modules that implement functionality that makes that application unique.
User interface management
Data and presentation management
The Code Editor
Setting management
Cross-platform
Wizards and code generation and management tools
The Wizard framework
Configuration management
Storage management

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Support for Ant, CVS and other version control systems
Visual design tools"

You can download it here: http://www.netbeans.org/products/index.html

- **Question:** What is jvmstat?

Answer: The jvmstat technology adds light weight performance and configuration instrumentation to the HotSpot JVM and provides a set of monitoring APIs and tools for monitoring the performance of the HotSpot JVM in production environments. The instrumentation is designed such that it is 'always on', yet has negligible performance impact. The monitoring interfaces added to the HotSpot JVM are proprietary and may or may not be supported in future versions of the HotSpot JVM.

Tools include a general purpose command line tool called jvmstat, a java process list tool called jvmps, and a generational heap visualization tool called visualgc. Remote monitoring is supported via an RMI server application.

The instrumented HotSpot JVM shipped with J2SE 1.4.1 as a technology preview. The instrumentation is disabled by default, but can be easily enabled through a command line argument. The tools included with this distribution can only monitor Java applications running on the HotSpot 1.4.1 JVM as distributed with Sun's J2SE 1.4.1 product.

The jvmstat feature includes a number of tools:

jvmstat - A tool to sample and log instrumentation

jvmps - A tool providing a simple Java process listing

visualgc - A tool to graphically display instrumentation

perfagent - A server providing remote access to instrumentation

tips is based on
http://developers.sun.com/dev/coolstuff/jvmstat/

- **Question:** Do you know any Java API to access Windows registry?

Answer: I know just one - JNIRegistry.
The com.ice.jni.registry package is a Java native interface for the Windows Registry API. This allows Java program to access, modify, and export Windows Registry resources.

The package will work only with Java 1.1 and greater, and uses the Javasoft native interface, not the Netscape interface. The package also includes a DLL that implements the interface. The package has been used with JDK1.2, and JDK1.3, as well as JDK1.1.8.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The package includes the pre-built DLL (debug and release), source code (both the Java and the DLL's C code), as well as the compiled Java classes.

You can download it here:
http://www.trustice.com/java/jnireg/

Good Link here:

I found most full table of existing Java obfuscators on http://proguard.sourceforge.net/ by Eric Lafortune. I am very thankful to him for permission to republish the page here.

"There are quite a few Java class file shrinkers and obfuscators out there. Users of ProGuard tell me it compares with the best of them. However, you may want to check that out yourself.

This is a list of the programs of which I'm aware. Obviously, I've never personally tested all of them. Note that many programs, even commercial ones, have been abandoned. Please drop me a note if you know of any other shrinkers or obfuscators, or if some information provided below is incorrect.

Author/Company	Program		Obfuscation	License
Eric Lafortune	ProGuard	x	x	Free (GPL)
Jochen Hoenicke	Jode	x	x	Free (GPL)
Hidetoshi Ohuchi	Jarg	x	x	Free (BSD)
NQ4	Joga	x	x	Free (no source)
Alexander Shvets	CafeBabe	x	x	Free
Nate Nystrom	Bloat	x		Free
RiggsHill Software	GenJar	x		Free (Apache)
RetroLogic	RetroGuard		x	Free (LPL)
Thorsten Heit	JavaGuard		x	Free (LGPL)
Patrick Mueller	Mwobfu		x	Free (GPL)
Bajie	JCMP		x	Free
yWorks	yGuard		x	Free
Dr. Java	Marvin Obfuscator		x	Free (no source)
IBM AlphaWorks	JAX	X	x	Commercial
PreEmptive	DashOPro	x	x	Commercial
Zelix	KlassMaster	x	x	Commercial
S5 Systems	jPrsto	x	x	Commercial
CodingArt	CodeShield	x	x	Commercial
Plumb Design	Condensity	x	x	Commercial
Eastridge Technology	Jshrink	x	x	Commercial
Helseth	JObfuscator	x	x	Commercial
LeeSoftware	Smokescreen Obfuscator	x	x	Commercial
Vega Technologies	JZipper	x	x	Commercial

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Sergey Sverdlov	J.Class Optimizer	x		Commercial
Force 5	JCloak		x	Commercial
WingSoft	WingGuard		x	Commercial
JAMM Consulting	ObfuscatePro		x	Commercial
2LKit	2LKit Obfuscator		x	Commercial
Duckware	Jobfuscate		x	Commercial
JProof	JProof		x	Commercial
4Fang	JMix		x	Commercial
GITS	Blurfuscator		x	Commercial
JDevelop	JSCO		x	Commercial
Alain Moran	flmObf		x	Free (BSD)
Vasile Calmatui	VasObfuLite		x	Free
Markus Jansn	Jopt	x	x	(disappeared?)
Eron Jokipii	Jobe		x	(disappeared?)
JRC	DeCaf		x	(disappeared?)
4th Pass	SourceGuard	x	x	Commercial (discontinued?)
HashJava	HashJava		x	Commercial (disappeared?)

All trademarks are property of their respective holders."

by Eric Lafortune

- **Question:** Is there any way in java we can view the bytecodes generated by a piece of code?

  Answer: Run this command in directory where you have your MyClass.class file:

  javap -c -classpath . MyClass

- **Question:** Does anyone use any tools to get only the classes that an application requires from rt.jar and other JDK jar files? We are looking to optimize an apps deployment.

  Answer: If you mean distributing the modified rt.jar to end users, this is prohibited by Sun's license. Here is Section 2 of the Supplemental License Terms from the JRE 1.4.0_02:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

"2. License to Distribute Software. Subject to the terms and conditions of this Agreement, including, but not limited to Section 4 (Java Technology Restrictions) of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license to reproduce and distribute the Software, provided that (i) you distribute the Software complete and unmodified (unless otherwise specified in the applicable README file) and only bundled as part of, ..."

Possible solutions:

Try to write to use the Java 1.1 standard. Most Windows PCs out there have some form of 1.1 installed (the Microsoft VM is roughly 1.1.4 equivalent). No swing (or you need to distribute a ~1 Mb swingall.jar file), and no Java 2.

(b) If you need a Java 2 - try to use the oldest version that you're comfortable with in your <EMBED>/<OBJECT> tag. Older versions have smaller size :-)
But you need to inform customers about it.

Ł **Question:** What is Java Shared Data Toolkit and how can I use it?

Answer: The Java Shared Data Toolkit software is a development library that allows developers to easily add collaboration features to applets and applications written in the Java programming language.

Enterprise developers can use the Java Shared Data Toolkit software to create network-centric applications, such as shared whiteboards or chat environments. It can also be used for remote presentations, shared simulations, and to easily distribute data for enhanced group workflow.

The Java Shared Data Toolkit complements Sun's rich suite of Java multimedia technologies by allowing them to be incorporated into "sessions" created and managed with the Java Shared Data Toolkit.

The Java Shared Data Toolkit software features:

- Compliance with Sun's 100% Pure JavaTM certification standards: the Java Shared Data Toolkit can be used in any Java

Compatible TM environment.

- Sample programs: provides examples of a chat environment, a shared whiteboard, a networked game, a stock quote viewer, an internet phone, a midi synthesizer jam session, a collaborative web browser and a sound server

- The ability to work through Internet firewalls.

- Robust, multi-threaded code: a substantial timesaver for developers.

- Small footprint: quick and easy to download.

This tip is based on SUN's information

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** What is difference between ordinary Java archive file and executable one?

  Answer: The difference between an ordinary jar file and executable is that executable jar file you run by double clicking and the runnable jar file contains a manifest file with:

  Main-Class: attribute.

  You also have to associate java.exe program with jar files.

- **Question:** I do not like to write JavaDocs, I need some tool to help me...

  Answer: I like DocWiz.
  DocWiz allows you to add JavaDoc comments to your source code easily. With DocWiz, there's no need to tediously hand-format JavaDoc comments, adding tags and comment structures for each method.

  Download the tool from here:
  http://www.mindspring.com/~chroma/docwiz/index.html

Question: I recently found that zip and jar classes were enhanced with two more flags with OPEN_READ an OPEN_DELETE. I do not understand why we need them. Could not we read or delete zip/jar files before?

Answer: SUN says the next: "The previous implementation of caching downloaded Jar files created problems for long-running server application and for RMI. Each open JarURLConnection creates a temporary local file that holds Jar-file date, and a JarFile object is created on top of the temporary file. Because the open file handle/descriptor on each temporary file was never closed, the files could quickly take up disk space for long-running server applications. The new API provides a solution for this problem by supporting a new "delete-on-close" mode for opening Zip and Jar files."

- **Question:** I need to do a lot of technical computing in my Java programs. Could you advice me, where to find free libraries to do it in most efficient way.
  I do not want to reinvent a wheel, at least now :-)

  Answer: Ok, there is one collection of Open Source Libraries for High Performance Scientific and Technical Computing in Java - The Colt Distribution.

  This distribution consists of several free Java libraries, for user convenience bundled under one single uniform umbrella. Namely the Colt library, the Jet library, the JAL library, the RngPack library, the CoreJava library, the Concurrent library and the VNI library.

  The Colt library provides fundamental general-purpose data structures optimized for numerical data, such as resizable arrays, dense and sparse matrices (multi-dimensional arrays), linear algebra, associative containers and buffer management.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The Jet library contains mathematical and statistical tools for data analysis, powerful histogramming functionality, Random Number Generators and Distributions useful for (event) simulations, and more.

The JAL library, a partial port of the C++ Standard Template Library developed by Silicon Graphics, contains a wide range of efficiently coded general-purpose algorithms on arrays.

The RngPack library contains random number generators.

The CoreJava library contains C-like print formatting.

The Concurrent library contains standardized, efficient utility classes commonly encountered in parallel & concurrent programming.

Finally, the VNI library contains mathematical functions and complex numbers.

And more can be found here:
http://tilde-hoschek.home.cern.ch/~hoschek/colt/index.htm

Ł **Question:** I am starting with Java now...
Could you point me some simple tool that can check for:
-unused methods, class usage, check my style...

Answer: Checkstyle - development tool to help programmers write Java code that adheres to a coding standard. It automates the process of checking Java code to spare humans of this boring (but important) task. This makes it ideal for projects that want to enforce a coding standard.

The things that Checkstyle can check for are:

- Unused or duplicate import statements.
- Javadoc comments are defined for class, interface, variable and method declarations. Note: This can be made to check only public and protected declarations.
- Javadoc tags for a method match the actual code.
- Detect instantiations of classes that should not be instantiated (e.g. java.lang.Boolean).
- Content of try, catch and finally blocks.
- Enforce line wrapping on operators.
- Presence of to-do comments.

And much more....
http://checkstyle.sourceforge.net/

IntelliJ IDEA is a full-featured Java IDE with a high level of usability and outstanding advanced code editing and refactoring support.

Its unique and best-of-breed features make it indispensable for both client and server-side application development.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Check it out yourself by looking through the features, downloading and trying this really smart and intelligent Java IDE.

You can take evolutional copy here (fully functional):
http://www.intellij.com/idea/

- **Question:** I like JBuilder very much. Just one thing - mouse wheel is not supported...

  Is it possible to fix it?

  Answer: Yes, it is possible. Please take a plugin here:
  http://czerwonykapturek.tripod.com/jbWheel/jbwheel.html

  JbWheel is an OpenTool that enables WheelMouse in JBuilder. This tool uses a native Windows DLL library so it won't work under other systems.

- **Question:** How can I install my Java program as a NT service?

  Answer: Use JavaService. The JavaService is free software. It is distributed using a BSD style license JavaService is a Win32 executable that allows any Java class to be run as an NT service. It uses JNI's invocation API to create a Java Virtual Machine using any JNI compliant Java implementation. JavaService can call the typical entry method:

  static void main(String[])

  in order to start the service, or any other method you choose. In order to stop the service it can call another method or simply terminate the JVM.

  Please take it here:

  http://www.alexandriasc.com/software/JavaService/index.html

- **Question:** How can I check CLASSPATH variable on my OS?

  Answer: Open Command prompt in Windows and type: echo %CLASSPATH%

  In Linux, Unix or MAC OSX in any shell: echo $CLASSPATH

- **Question:** Which Java library can convert psd format (Photoshop Image)? Is it free?

  Answer: ImageMagick Studio, a non-profit organization dedicated to making software imaging solutions freely available.
  ImageMagick TM 5.4.9 is a robust collection of tools and libraries offered under a usage license to read, write, and manipulate an image in many image formats (over 87 major formats) including popular formats like TIFF, JPEG, PNG, PDF, PhotoCD, and GIF. With ImageMagick you can create images dynamically, making it suitable

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

for Web applications. You can also resize, rotate, sharpen, color reduce, or add special effects to an image and save your completed work in the same or differing image format. Image processing operations are available from the command line, as well as through C, C++, Perl, or Java programming interfaces.
http://www.imagemagick.org/

- **Question:** Can Java communicate with ActiveX objects?

  Answer: Yes, if you use Bridge2Java from IBM. Bridge2Java is a tool that allows Java programs to communicate with ActiveX objects. It allows easy integration of ActiveX objects into a Java Environment. Using the Java Native Interface and COM technology, Bridge2Java allows an ActiveX object to be treated just like a Java object.
  How does it work?
  Using Bridge2Java requires simply running a proxy generating tool that creates Java proxies from the ActiveX control's typelib. These proxies can then be used to allow a Java program to communicate with the ActiveX object.

  Free trial 90 days...
  http://www.alphaworks.ibm.com/tech/bridge2java

-

-

11. Job, fun

-

- - "Have you heard about the object-oriented way to become wealthy?"
  - "No..."
  - "Inheritance."
- **Question:** Please clarify my confusion about Java versions. I hear Java 2 but JDK is 1.2

  Now we are going to use JDK 1.3 on the job and I said that this is Java 3 and probably soon when JDK 1.4 will be released we will use Java 4.

  My chief said me that I am wrong and he never heard about Java 4. He insists on it, but cannot explain why.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Maybe because of he is chief? Please explain...

Answer: He is right and not because of the fact that he is chief :-)

Although exist one old set of rules:

1. Chief is always right!
2. If chief is not right please see the rule number 1.

I think we have to wait a few years more until Java 3 comes to us.
It is SUN's numbering policy. All JDKs that have numbers 1.2, 1.3, 1.4 are Java 2 JDKs...

Difficult to understand but easy to remember...
--
AP. (J.A.)

- **Question:** Is text file binary?

Joke, not **Answer:** 8-)

Yes, ALL FILES are binary, even text files... Because a file is a series of bits...

And bits are binary...

- **Question:** Hi, I am learning java and hoping to take the certification test this April. How important is it to gain this certification? And would employers be interested?

Answer:

1. Look at the job ads... do they ask for it?
2. If you are an interviewer with two identical candidates, but one had this. Who would you employ?

- **Question:** ...However I am having a difficult time finding a job.

I have been studying Java since December and passed the SCJP a few weeks ago. However I am having a difficult time finding a job. I have found that most of the positions require a good deal of experience first. Also, even the entry level positions have not responded to my resume.

Two questions:

Any ideas on how I can make it look better?
and even more importantly,
What should I begin learning next?

(There are so many advanced concepts, JavaBeans, Swing, JSP, EJB etc.. that I am a little bit lost).

Answer 1: I graduated with a bachelor's degree in college just a few months ago and I am currently working for a company that uses Java quite heavily. I was lucky that I got onto the staff that I am, but it wasn't all luck. Companies are looking for

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

people who can not only program, but also communicate well and find the answers to questions on their own. Personally, if I were an employer that saw you completed the tests on your own, that would tell me that you have self-motivation and the yearn to learn: a definite plus considering how fast technology changes.

Bottom line is this: getting a job is tough, no matter what level experience you have. The main thing to keep in mind is that time is on your side. Your patience will win in the end.
--
by RyanDecker

- **Question:** ...However I am having a difficult time finding a job. Part 2

  Answer 2: Try with different government institutes (i.e. local, municipal, and federal) when you are first starting out. Usually the more experienced programmers in the government go to the private industries after a few years and the government is looking for people to replace the ones who have left and this is usually a good place to start.

  I hope this helps. Good luck in your search.
  --
  Robert

- **Question:** I would like to know if the small programs that are attached in mail (like the small games for example) are usually coded in java or in any other language.

  Answer: Usually not. Usually they're viruses.
  --
  Nils O. SelAsdal

- **Question:** An ordinary guy of ordinary intelligence, with some experience of html, do I stand a chance of learning java or do I need to be of an above average genius. Any feedback would be welcome!

  Answer: Java is extremely easy to learn...but will take a lifetime to be a master. Intelligence is not an issue.
  --
  drichard_007

- **Question:** I have recently passed my Java Programmer Certification in the UK. Is there a standard logo I can use for my homepage & resume?

  I have contacted Sun with this matter, but have received no reply.

  Answer: You will receive information concerning how to get the logo, along with a "logo use agreement" from Sun in your SCJP package. As I recall, according to the agreement that you have to sign and send back to Sun before you are given the URL.

  Before, when you get old agreement, you agreed not to place the logo on a web page or on your resume.

  With new agreement you may use the logo in your resume. The new agreement is

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

now the standard. All the candidates who signed an old agreement will be under the rules of the new agreement.
--
Clyde, Tym Tyler Here

12. Miscellaneous

- **Question:** What is the difference between Java and Microsoft Visual J++? Is there any specific difference? Please explain me I am novice to this field Part1

  Answer: VJ++ was designed as a "spoiler" product -- Microsoft created some deliberate subtle incompatibilities so that when people wrote VJ++ code it would only run on Windows. This is not my opinion. This came out in evidence in Microsoft's trial for anti-competitive behavior.

  In email they said things like they were going to "pollute" Java, and "piss on" the Swing libraries.

  Some of the incompatibilities were in removing some standard fields from the system libraries. Others were adding some fields. They also added some differences to the language itself, in the way event handlers were registered.

  When Sun found out about Microsoft's attempt at sabotage, they cut-off all code deliveries to Microsoft, and sued Microsoft. That lawsuit just ended with the payment of $20M by Microsoft to Sun.

  So Microsoft VJ++ is several releases out of date, and does not have many of the most important libraries, such as RMI and beans.

  Bottom line: even though it has a nice GUI IDE, if you want to program in Java, you are better off avoiding VJ++, and using any of the free IDEs mentioned in the Java FAQ.

  by Peter van der Linden http://www.afu.com

  P.S by John: please read second part of this tip tomorrow and you will see you can use it with JDK1.3!

- **Question:** Has anyone found a way to produce a MS Word file with Java?

  Answer: yes and it is free! Java bean word processing. It is an interface to control Microsoft® Word by java applications (includes German Version)

- Would you like to load Microsoft® Word, to create a new document and to serve bookmarks by your java application? There is a free java class for you here! http://www.must.de/default.html?Javactpe.htm

- **Question:** What is the difference between Java and Microsoft Visual J++? Is there any specific difference? Please explain me I am novice to this field Part2

  Answer 2: Microsoft Visual J++ is a Java IDE for editing, compiling, and debugging Java source code. It also provides GUI editing tools that generate code for you.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Visual J++ added some extensions to the pure Java language that you can easily disable.

The latest version of Visual J++ is 6.0. It supports JDK 1.1.
In order to use JDK 1.2, JDK 1.3, or beyond you must follow the procedure below:

Modify the CLASSPATH in the registry. Run RegEdit.exe and locate the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Java VM
and modify the CLASSPATH string to include the rt.jar file from you JDK distribution.

For example:

c:\Program Files\JavaSoft\JRE\1.3\lib\rt.jar

needs to be added to the end of the CLASSPATH string, and don't forget the semi-colon separating each entry in the string. Also note the "." at the end of the CLASSPATH string.
--
Roger L. Cauvin
http://www.thegym.net/rcauvin

- **Question:** I developed a package that I am going to sell as a jar file. I want that my package can not extended externally in form of some additional class files or another jar file. I want that only classes that are in my jar file will be considered "real" classes that belong to my package.

  Excuse me for such difficult question (in terms of English), I am from Spain

  Answer: That's right, exist some possibility to "crack" your code with some external "home made extensions" and make your application believe that everything is still ok!

  For example, exist an application and it uses some time based licenses that classes from within that jar file check...

  Let say, I do not want to pay a license fee. I know that jar file (check license date and payment) lays somewhere from where I can read but can not modify or delete.

  What can I do? I can write some classes that will violate license functionality and declare them "members" of that package and will "feed" it to my application in classpath.

  Yes, you can forbid to all other classes that are outside of jar file and "claim" that they are another part of the same package to be used by an applet or application. You must "seal" your jar file.

  Even though this feature was introduced to use versioning, you can use it top protect your application

  In original SUN document (below) I found:
  "JAR files and packages can be optionally sealed, so that an optional package or package can enforce consistency within a version.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

A package sealed within a JAR specifies that all classes defined in that package must originate from the same JAR. Otherwise, a SecurityException is thrown.

A sealed JAR specifies that all packages defined by that JAR are sealed unless overridden specifically for a package.

A sealed package is specified via the manifest attribute, Sealed, whose value is true or false (case irrelevant). For example,

Name: javax/servlet/internal/
Sealed: true

specifies that the javax.servlet.internal package is sealed, and that all classes in that package must be loaded from the same JAR file.

If this attribute is missing, the package sealing attribute is that of the containing JAR file.

A sealed JAR is specified via the same manifest header, Sealed, with the value again of either true or false.

For example,
Sealed: true

specifies that all packages in this archive are sealed unless explicitly overridden for a particular package with the Sealed attribute in a manifest entry.

If this attribute is missing, the module is assumed to not be sealed, for backwards compatibility. The system then defaults to examining package headers for sealing information.

Package sealing is also important for security, because it restricts access to package-protected members to only those classes defined in the package that originated from the same JAR file.

Package sealing is checked for installed as well as downloaded optional packages, and will result in a SecurityException if violated. Also, the null package is not sealable, so classes that are to be sealed must be placed in their own packages."

- **Question:** I can use URLConnection class to get a file from another site. Can I "on the fly" to get somehow files from within of jar file through an URL connection?

  Answer: Yes, you need to use quite old class from Java API: JarURLConnection. "The syntax of a JAR URL is:

  jar:<url>!/{entry}
  for example:

  jar:http://www.foo.com/bar/baz.jar!/COM/foo/Quux.class

  Jar URLs should be used to refer to a JAR file or entries in a JAR file. The example above is a JAR URL which refers to a JAR entry. If the entry name is omitted, the URL refers to the whole JAR file: jar:http://www.foo.com/bar/baz.jar!/

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Users should cast the generic URLConnection to a JarURLConnection when they know that the URL they created is a JAR URL, and they need JAR-specific functionality. For example:

```
URL url = new URL("jar:file:/home/duke/duke.jar!/");

JarURLConnection jarConnection = (JarURLConnection)url.openConnection();
Manifest manifest = jarConnection.getManifest();
```

Examples:

A Jar entry
jar:http://www.foo.com/bar/baz.jar!/COM/foo/Quux.class

A Jar file
jar:http://www.foo.com/bar/baz.jar!/

A Jar directory
jar:http://www.foo.com/bar/baz.jar!/COM/foo/

!/ is refered to as the separator"

From Java API for JarURLConnection class. Read more examples in API.

- **Question:** Can I convert my Delphi code to Java?

Answer: Yes you can with program Delphi2J, see details at:
http://www.javadelphi.com/

Question: Do exist non SUN's, freeware implementations of essential libraries for supporting the Java language?

Answer: GNU Classpath is a set of essential libraries for supporting the Java language. Our goal is to provide a free replacement for Sun's proprietary class libraries so that the GNU system can run programs written in the Java language.

GNU Classpath 1.0 will be fully compatible with Java 1.1 and largely compatible with Java 2..
More info here: http://www.gnu.org/software/classpath/

- **Question:** I want to use COM component from Java. How can I get an access to the COM interfaces?

Answer: J-Integra is a bi-directional pure Java-COM bridge.

J-Integra allows developers to work in their respective environments, while having the benefit of accessing COM components from Java, and Java components from COM. The ultimate benefit is that developers can save time by leveraging existing e-business software code, deliver best-in-class software solutions, and avoid learning new programming environments:

- Pure Java DCOM implementation
- Can be used with any Java Virtual Machine (JVM), running on any platform
- Access COM from Java and vice versa

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- High-speed native mode for Windows platforms
- Easy to use with lots of step-by-step examples
- Zero client and server install options - no JVM or other additional software required on the COM platform

It is not free. I think this kind of product requires much knowledge to develop it. See more at http://www.linar.com/

Ł **Question:** Is it something new in J2SE 1.4.1?

Answer: I would like to say about next things that I was glad to read about:

1. Over 2000 bug fixes!!!

2. Two new garbage collectors have been added to improve application performance.

Concurrent Mark and Sweep Collector.
This collector executes mostly concurrently with the application. It trades the utilization of processing power that would otherwise be available to the application for shorter garbage collection pause times.

Parallel Garbage Collector.
This collector enables garbage collection to occur on multiple threads for better performance on multiprocessor machines.

P.S. A few flags were added to command line to manage those GSs...

3. The Sun SecureRandom implementation now also makes use of an operating system-provided entropy source on Windows platforms, which can improve the start up time of cryptographic applications considerably.

4. A deadlock detection utility has been added to the Java HotSpot VM. The utility is invoked by a ctrl+\ (on Linux or the Solaris Operating Environment) or a ctrl-break (on Microsoft Windows) on the command line while an application is running. The utility detects Java-platform-level deadlocks, including locking done from the Java Native Interface (JNI), the Java Virtual Machine Profiler Interface (JVMPI), and Java Virtual Machine Debug Interface (JVMDI).

When invoked, the utility displays a thread dump to standard out and indicates any Java-platform-level deadlocks it detects.

Refer to this sample output. If the application is deadlocked because two or more threads are involved in a cycle to acquire monitors, then the list of such threads and monitors involved in the deadlocks are displayed. Note, however, that this will not find deadlocks involving threads waiting on monitors on which no signal will be forthcoming.

Ł **Question:** Can the JavaHelp software run in a non-Java application, such as a C++ application?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Yes, the JavaHelp software can provide help for applications written in languages other than the Java language. This is possible by through the use of a help server.

Answer based on:

JAVAHELP SOFTWARE FAQ

Question: Where is MS Java Virtual Machine stored in IE? I would like to take them away... And use only original plugin from SUN...

Answer: There are two places:

windows\system32\msjava.dll and

\windows\java

I am not sure that you delete everything from \windows\java since there a lot of SUN packages as well.

- **Question:** I was considering an installation of java application on a new computer without the JVM installed, and wondered if a "silent" (in background, without questions) install of the JVM is possible...

Answer: Yes, it possible.

Java 2 Platform Runtime Environment 1.3 installations are built using the InstallShield 5.1 and InstallShield PackageForTheWeb products. These products contain built-in support for silent or unattended installations.

A normal (non-silent) installation receives the necessary input from the user in the form of responses to dialog boxes. However, a silent installation does not prompt the user for input and must get its user input from a different source. That source is the InstallShield Silent Response File (.iss file). A response file contains information similar to that which an end user would enter as responses to dialog boxes when running a normal setup. InstallShield reads the necessary input from the response file at run time while performing a silent installation.

- **Question:** What can I do if some programming language has no methods to interact with Java program? I need to pass some info from Perl, PHP ... program to Java program...

Answer: Best solutions which will work with most of languages are:

- make network connections, or if there is no networking functionality
- write/read information into/from file.

In both cases you need to implement your own protocol. In case with file you can use lastModified() to see when some info was sent from another program. If it is different from previous one it means that a new info was sent... Then just read it and

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

use according to your protocol.

- **Question:** Can I convert Visual Basic code to Java?

  Answer: Probably it is possible, at least creators of site http://www.blackdirt.com/ offer you to try your examples online...

- **Question:** What is an asynchronous architecture?

  Answer: In asynchronous architecture a program does not wait for return immediately. The program continues to do another staff and when does not care when it will happen.

  Simple example: modeless dialog. You can do something else and return later to that dialog, change something and press “Apply“ button.

  Synchronous example: modal dialog call, the procedure call it will not continue until the model dialog returns. For example, when you try to delete files you will be asked: “Are you sure that you want to delete file(s)?” You can not continue until you press “Yes”, “No” or “Cancel”.

- **Question:** I want to know, can Java be run in DOS?
  Not Visual J++ but just Java, can it run in DOS?
  If so is the language is the same? I have been told that java can run in DOS but I just want to verify...

  Answer: Java technology relies on files with longer names than the old DOS limits, so if you are running older (pre-Microsoft Windows 95) versions of DOS, Java technology will not work. If you are running a version of DOS (such as the one that comes with Windows 95) that allows long file names, you should not have any problems.

  Note that older versions of WinZip do not support long file names, even if the installed version of DOS does. You can get a free upgrade of WinZip with support for long file names from their web site.

  source: http://java.sun.com/products/jdk/faq.html#E1

- **Question:** I am just wondering. What is the difference between java and java2?

  Answer: Java2 is really just another name for JRE/JDK1.2 and higher. Various things improved dramatically between JDK1.1 and JDK1.2, mainly in the libraries - things like the collection classes suddenly appeared.

  The extensions mechanism was also introduced, and I suspect the security policy was firmed up (I've never needed to investigate that, fortunately).

- **Question:** I am curious to know why Sun seems to ship two virtual machines (VMs) with their software development kit (SDK, a.k.a. JDK). I am talking in particular about JDK 2 v1.3 for win32, although this question may apply to other versions.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Here is where the VMs are on my installation:
c:\Program Files\jdk1.3\bin\java.exe
c:\Program Files\jdk1.3\jre\bin\java.exe

Incidentally, it also appears that there are lots of libraries in the jre\bin subdirectory that are not present in the other bin directory.

Both files are 21k on disk, and it seems unlikely that either of them are actually a true VM.

Answer: The EXE files aren't the VMs, they're just wrapper that invoke the VM. The actual VM files are called JVM.DLL, and you'll find them in:

jre\bin\classic (basic, interpreted VM).
jre\bin\hotspot (the real thing).

If you invoke java.exe, you'll get hotspot by default. Use "java -classic" to get the old VM.

The main difference between them is that when you run java then you also get access to the java compiler and some extra "developer" stuff.

When running JRE you get a more streamlined version. Notice that it is possible to download just the JRE, which is a lot smaller than the full JDK.
--
Kenny

- **Question:** How to set the default memory limit of java virtual machine in a java application?

Answer 1:
java -Xms16m -Xmx32m MainClassName

here: -Xms16m => 16meg initial memory allocation
-Xmx32m => 32meg max memory allocation

Answer 2: Run your Java program with -mx<number> switch like this:

java -mx128m ClassName

This for example will set maximum memory allocation pool to 128MB.

- **Question:** Do the classes in java.util.zip handle password-encrypted zip files? I've looked through the API, and I don't see any mention of it.

Answer: No, they don't. But Zip's built-in encryption isn't safe anyway by today's standards.

- **Question:** I have a ZIP file and I want to check the file type of its entries WITHOUT unzipping it. Anyone knows how to do it?

Answer: You can't do this, unless knowing the file suffix is enough information for

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

you. Then you can simply list the file names using the ZipFile class "entries ()" method.

If you cannot determine the file type without examining the file itself, then obviously you must read the file, i.e. unpack the ZIP file.
--
Paul Lutus, www.arachnoid.com

- **Question:** I want to compile a source file(.java) and then run it (.class) from inside another java program. How can I do that?

  Answer: Make sure you place tools.jar on your class path.

  If using JDK < 1.3, then use

  public static int sun.tools.javac.Main.compile(String[] args);

  if using JDK >= 1.3 then use

  public static int com.sun.tools.javac.Main.compile(String[] args);

  then if the result code indicates all is well (==0) the just go ahead and load the resulting class file, probably in a new class loader.
  --
  Neal M Gafter

- **Question:** How do I make java application installable? I have written a Java application and have the .class files in one location. How do I make it installable? Is it possible to convert it to an executable file?

  Answer 1: just create a batch file or a .sc file (if on UNIX)
  Answer 2: Or check http://installshield.com/

  They have special edition for Java which lets to write additional interface for configuring of program parameters during installation.

- **Question:** I've got a Java application that I need to install on my future customer's computers but this is the first program that I've ever tried to sell to anybody and I want to keep my costs down.

  Is there something out there that installs the JRE, if it needs to, and then my program?

  I know of InstallShield, InstallAnywhere and the like but I'm hoping to find something a lot cheaper since I'm just a small time operator (who may not sell even one of his programs).

  Answer:

  1. ZipCentral is a free Windows app that will create a zip file, or a self-extracting executable. It has the ability to execute a BAT file after unzipping your files. You can download a copy from

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

http://zipcentral.iscool.net/

2. Sun's Java Web Start (the price is right - free):

http://java.sun.com/products/javawebstart/index.html

Java TM Web Start -- a new application-deployment technology -- gives you the power to launch full-featured applications with a single click from your Web browser. You can now download and launch applications, such as a complete spreadsheet program or an Internet chat client, without going through complicated installation procedures

3. Zerog's Now! http://www.zerog.com/downloads_01.html

There are no restrictions on the use of free products:

InstallAnywhere Now! and PowerUpdate Now!. Now! 4.01 is free, installs a JRE

- **Question:** I need to know how to run Multiple JVMs on one machine. As far as I know if I run 2 different Java Programs on one machine at the same time, both of these programs use the SAME JVM. Is this statement right?

  Answer: No. Each invocation of the 'java' command creates a new and separate JVM, at least in those JVMs based on the Sun code, and getting them to cooperate on a shared Java task is not automatic, and definitely non-trivial.

- **Question:** I am totally confused about the differences between the SDK, JDK and IDE products.

  I am brand new to the world of Java and am most interested in learning the language. However, I am confused about the differences between the SDK, JDK and IDE products. I have gone to the Sun site and even some of the IDE vendor sites and I have yet to find something that tells me what the differences are between the three and which of the three I need to program in Java.

  I want to program using some form of IDE but do I need to separately download and install a SDK and/or JDK? Sun needs to improve their documentation for us new to the Java environment.

  Answer 1: IDE is an acronym for _I_ntegrated _D_evelopment _E_nvironment. These products are the one stop shops for coding, running and debugging your code. Often these will include GUI based drag and drop form designers and "wizards" for the shells of common forms of code (Application, Applet, etc.) JBuilder is an IDE.

  The IDE may stand on its own, or it may act as a front end for a JDK.
  JDK is _J_ava _D_esign _K_it. A JDK is a command line based interface to the JVM, plus the classes. You are responsible for your own editors, creating code for GUI elements, and all code. All of the IDE's I have reviewed personally come with JDK or their own vendor's equivalent (JVM and class libraries). Some IDE's are capable of a sort of "upgrading" by downloading the latest JDK from Sun (JBuilder for example).

- **Question:** I am totally confused about the differences between the SDK, JDK and IDE products. Part 2

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer 2: If you want to write a Java program using any editor and not an IDE then you would want to download the JDK. It will let you compile and run Java programs from the command line (like a DOS window).

JDK stands for Java Development Kit and SDK stands for Standard Development Kit.

Java comes in three versions - Standard, Enterprise, and Micro editions. JDK could be any one of the three. SDK is the standard one - this is the one most people use. If you want an IDE they typically come with a JDK so all you would need to do there is download the IDE and start using it.

- **Question:** How does a java application stored within a jar file reference/edit/read other files (like .txt, or data files,) that are also within the jar file?

  Answer: Classes located in a JAR archive are loaded via a class loader whose purpose is to load classes form JAR archives. This ClassLoader implements the getResource and getResourceAsStream methods to retrieve files from the JAR file. So you can take any class from the JAR and say ClassName.class.getClassLoader().getResource("fname"); to get the resource and use it.

- **Question:** I want to keep my java GUI always on the top of any other desktop application. Any idea?

  Answer: Spawn a thread that knows about the parent Window, and every X milliseconds, executes the toFront () command of that window. Just remember to execute it using SwingUtilities.invokeLater (), and don't let your users launch two apps, unless you enjoy screen lockup.

- **Question:** I am beginner in Java and know that Java program runs on Java Virtual Machine. But how does Java works with file systems? They are real. How Java does solve this problem? For example if I want to modify ACL (Access list) on UNIX.

  There are no ACLs in Windows. No such method in API as well.
  Does it mean that file system of Java is quite limited?

  Answer: Yes, it is limited in some sense... Java supposed to run everywhere. "Everywhere" actually means that SUN limited file system functionality by common (for most OSs) methods. It is strength of Java and weakness as well.

  Since Java 1.2 we got at least some functions like: setLastModified() similar to touch() function in UNIX.

  Or listRoots() returns the list of all disks available in the system. It is important for Windows systems (On Unix all partitions are mounted to root "/." and has no names like in Win: A:\, C:\)

  I hope in future we will get even more.
  --
  Luis

- **Question:** Can a java application be run of a CD without installing anything (i.e. runtime, etc) on the target computer?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

I would like to put my application and hand it out as a demo, but I want to make it easy to view.

Answer 1: by Dale King The JRE was made so that it didn't need to be "installed". What I did in one case was to simply put the JRE into a jre folder in the same directory as my application then invoke it from that directory using:

jre\bin\jre.exe -cp MyJar.java MyClass

That was for JDK1.1 and you have to modify it slightly for Java 2. But this did not require any installation of environment variables to be set up. The JRE was smart enough to know how to get to its system classes relative to where the jre.exe file was located.

Answer 2: you could try a Java to native compiler.

- **Question:** I would like to know whether it is possible to test the memory, so as to avoid the OutOfMemoryError or whether it is possible to increase the amount of memory in the JRM.

Answer: You can get the total and available memory used by the VM by making two calls from the Runtime class:

```
Runtime runtime = Runtime.getRuntime();
long free = runtime.freeMemory(); //the available memory
long total = runtime.totalMemory(); // the total for the JVM
```

The amount returned be totalMemory() isn't that useful unless you specify how much memory your program will have from the beginning (if you don't, the JVM will just keep grabbing more until you run out).

You can set the initial and maximum memory from the command line:

java -Xms64m -Xmx64m name.of.Application

This will start your appplication with 64 megs initial and maximum memory.
--
Corey Wineman

- **Question:** I run some code in my program like this:

```
Runtime rt = Runtime.getRuntime();
rt.exec("C:/StartMyApp.bat");
```

I get very strange behavior: sometime it starts from first time, sometime I need to call it a few times and often it does not start at all after I run it once..

Answer: You need to modify your code since you execute your bat file in the MSDOS shell (Win95/98) or command line shell"cmd.exe" (NT).

You should use exec(String[] cmdarray) not exec(String cmd) method...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** What needs to be done to reduce size of a jar file? What optimization techniques to use on classes inside the jar file? What tools if any?

  Answer: A JAR file is a ZIP archive. You can influence its size by choosing the degree of compression you want to have. This is usually defined by a value between 0 (no compression) and 9 (maximum compression). Although JAR tool does not list a -9 switch, you might want to create compressed JARs with any ZIP tool like WinZip or the free Info-ZIP zip command line tool.

  The amount of reduction you get totally depends on the nature of your data.
  Note that if you use compression in your JAR file, loading goes slower (classes must be decompressed).

- **Question:** Is there any way to run code after the VM has been instructed to exit?

  Answer: In 1.3, you can use Runtime.addShutdownHook(Thread hook)

- **Question:** Where can I find Java --> Native code compilers?

  Answer: We just published a list of Java products that lets you do

  Java --> Native code compilation:

  http://javafaq.nu/java/staff/staff.shtml

- **Question:** I have a directory having class files arranged in package hierarchy. How can I make the executable of this whole directory?

  Any application available for that in Windows NT environment.

  Answer: Make a JAR file out of it and add a manifest file that indicates which main() method of which class must be called.
  Double-clicking this JAR file will run your application.

- **Question:** I'm interested in writing a little mp3 player in Java...

  I'm interested in writing a little mp3 player in java. I have an entirely different app right now that plays sound (wav files), and I substituted an mp3 file for one of the waves but it didn't work. Can anyone tell me if java even supports mp3 files?

  Answer: Go to the "Products & APIs" section of java.sun.com and look for JMF (Java Media Framework). It's a library that also supports reading MP3 files.

- **Question:** Are there any tools out there that will convert a program written in C to JAVA?

  Answer: Yes.

  C2J: http://www.novosoft-us.com/NS2B.nsf/w1/C2J

  C2J has successfully compiled itself as well as programs such as PGP and YACC. Obviously YMMV.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Based on C2J is a C++ to Java tool: http://sol.pace.edu/~tilevich/c2j.html
--
jim

- **Question:** Can we create DLLs in java If yes How?

  Answer: Unfortunately it is impossible. DLL stands for Dynamic Linking Library and has definite structure inside. DLL is a part of executable code and helps to make an application for Windows to be smaller. And more flexible. It is something like classes but compiled (Java class files are byte codes and JDK compiles them during the runtime...).

  In Java it is not possible to make an executable code. But with third party software Yes!

  See native compilers on our site. But I didn't hear about creating DLLs.

  So my answer is: in Java it is not possible, but with third party applications it is possible theoretically. Although I do not know any compilers that produce DLLs there are no limitations to do that....
  --
  John

- **Question:** Can someone tell me the difference between the JRE that comes with the J2SDK and the stand-alone JRE?

  Can someone tell me the difference between the JRE that comes with the J2SDK and the stand-alone JRE? When should I use which? I read the sun's web page and they said the JRE stand-alone package is for shipping your application, it doesn't come with a compiler nor debugger, but what does it really mean?

  Answer: Exactly that. The Java 2 SDK (a.k.a. JDK) is the JRE plus the compiler (javac) and debugger (jdb). The JRE is entirely sufficient to run a Java application (with a couple exceptions in situations where you call into the compiler or some such). It's entirely redistributable with your Java application if you've written an app in Java intended for an audience that may not have the JRE installed.
  --
  Chris Smith

- **Question:** Does anyone know of a java machine that will run from a 1.44Mb Floppy? I have an application I want to run from a dos 7.1 floppy disk.

  Answer: please check here:

  http://www.transvirtual.com/kaffe-features.htm
  there is written:

  "Efficiency is not just about execution speed, but Kaffe's JIT is quite speedy: it runs Java code only 30% slower than plain C. Such things as memory consumption of a JIT-enabled, graphical Kaffe are also important.

  Here we can execute a full system on a 4 MB DOS system, and the VM and

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

library footprint won't exceed 1 MB. Our complete source tree fits on a single 1.4 MB floppy.

This is what we mean by "efficiency".
Good enough for you?

- **Question:** Anybody know a good tool to distribute your made java classes with for windows platforms?

  I want to create a setup file that also verifies existence of java runtimes etc etc and creates a shortcut and that kinda things...I tried looking into Wise installation systems and InstallShield as well, but they are both not really build for this kinda things...

  Answer: You should try, InstallAnywhere by ZeroG (http://www.zerog.com)

  ZeroG has a free version call, "InstallAnywhere NOW". You should give it a try, its a really easy-to-use java installer. It has all the features you need, such as creating an Icon on your desktop, creating a folder under the Windows "Start" button, it can be set to have the user search for a JVM on their computer, and much more... (I sound like a commercial..)
  --
  JHig310336

- **Question:** I see all the time in the code examples that some Exception is caught but has empty body. Is it a good practice?

  Answer: No, it is not! Of course it lets to run program but does not treat the problem. It should be logged at least for further analysis. This "lazy programming" approach finally fools even the author.

  For example, in case of network connection: no Exception report, and of course no connection. If your program full of such Exception handlers you will just sit and guess what has happened.
  --
  AP. (J.A.)

- **Question:** Heap size limit!! I am running JVM from JDK 1.2 on Solaris 2.7 and I couldn't allocate the max heap size over 2G when I invoke the JVM.

  I have repetitive tasks that take 500m memory each to run, so I naturally want to run as many threads as possible. I figured out this 2G (-mx2047m) limit by trial and error but is there any way out of this? My workstation happen to have 2G physical memory, and the file size limit is 2G as well (from ulimit), are there any co-relation among those numbers?

  Answer: Yes, there is a relation: both result from limiting addressing space to what you can get with signed 32-bit ints for addresses:

  2^31 - 1 = 2 * 2^30 -1 = 2 * 1 GB -1 = 2 GB

  One of the interesting features of Java, is you could run the code with 64 bit addresses, and nothing would need to change in either the source code or the class

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

files. The only difference would be you could hold a lot more objects and stack frames before you blew virtual RAM.

Obviously you would need a different JVM or Hotspot.

Java never lets you discover how big references really are inside or how they are implemented, e.g. as pointers or as handles, or even the granularity of the addressability of your machine.

On a 32-bit OS it is a little bit hard to get 32+ bit memory space. How about you try a 64-bit Solaris 8?
answered by Michiel, Roedy Green,
JAVA GLOSSARY see http://www.mindprod.com/jgloss.html, Johnny Bravo

- **Question:** How can I format a diskette using Java? And, what's the way to get the diskette's size?

Answer: As far as I know: no. Formatting a disk is strongly system dependant, so you can hardly do this with java.

You *can* however start a new process, have it open a shell or any other kind of command processor your particular operating system uses, and issue the format command.

> And, what's the way to get the diskette's size?

Again system dependant.
--
Ansgar W. Konermann

- **Question:** I am a 17 year old in the JDK version 1.1. Is it worth upgrading and getting a later package, and learning swing?

If so, why?

Answer: Yes.

Java 1.3 provides many performance improvements and library additions which will make your life as a programmer easier.

Swing is "better" than the AWT in that because it is mostly written in Java - it has the advantage of the same look-and-feel across different platforms (which the AWT was crap at). The reference of the other learned poster to a mainframe is the common argument that Swing is slow. This _is_ true, however only if you build large applications and I doubt that at the moment you will.

In the future you also have the option of compiling your Swing classes to native code (viewed as a venerable evil by some, but a practical solution by others) or performing other optimization tricks that will see your GUI apps not only looking great and being cross-platform, but also performing well.

It's what we call in the trade a no-brainer. Time to get Swinging young chap.
--

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

pip

- **Question:** Is it possible to create a Jar file that is not unjarable?

  Or only unjarable on a certain domain/server? Is the jar.exe JDK-specific (I don't believe so)? Was I just asleep at the command line and imagining the whole thing?

  Answer: You could conceivably encrypt a jar with a password, but you wouldn't be able to start the application from that jar. You could have another jar that actually knows how to decrypt it and creates its own decrypting class loader. Of course your startup jar could be hacked allowing someone to figure out how to decrypt the jar. So once again, you can slow down the process and make it more painful but you can't make it impossible.

  To make this workable it would probably be a lot easier to encrypt the files within the jar rather than the jar itself, since you need random access to the jar file, but only sequential access to the files within. It is more difficult to write a good random access encryption scheme. This would allow you to unjar the files, but the files would be unintelligible. You might also apply a cipher to the file names in the jar so someone would not know whether a file was a class or a resource file.
  --
  Dale King

- **Question:** How does the system find the path to JDK 1.2 after I type in "java -version"?

  I installed the jdk1.2 on a NT system, also VisualCafe4.1 with jdk1.3 was installed on the same system. After typing in "java -version" in a DOS window, I always see java.exe 1.2 is invoked even though I couldn't find the path to JDK 1.2 from the system environment parameters. How does the system find the path to JDK 1.2 after I type in "java -version"?

  The reasons I ask this question because I want to invoke jdk1.3 under a DOS window without uninstall jdk1.2. I did add a path to jdk1.3 in system environment and reboot the system, but JDK 1.2's java.exe was still invoked after typing in "java -version" in a DOS window.

  Answer: Because when the JDK install kit placed the two programs java.exe and javaw.exe in the WINDIR and that's in the PATH. And these programs read the registry to find the JDK/JRE.

  If you placed jdk1.3\bin in the PATH before WINDIR, you should be fine. From your description, I guess you didn't, so you're still getting jdk1.2. Or you can directly execute jdk1.3\bin\java.exe which will work.

  Another trick I use to swap JDK's in an out is to place the directory named java\bin into the PATH and just rename the desired JDK to java, like this:

  c:\>dir j*

  Volume in drive C has no label
  Volume Serial Number is 07CF-0B17
  Directory of C:\

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
JDK12~1 2 <DIR> 03-19-00 1:57a jdk1.2.2
JDK13~1 0 <DIR> 12-21-00 1:42a jdk1.3.0
JDK11~1 8 <DIR> 06-16-00 2:29p jdk1.1.8
0 file(s) 0 bytes
3 dir(s) 16,252.02 MB free

c:\>PATH
PATH=c:\java\bin;c:\tools;c:\windows;c:\windows\command
c:\>ren jdk1.2.2 java
```

If I wanted to switch to JDK 1.3.0, I simply do:

```
c:\>ren java jdk1.2.2
c:\>ren jdk1.3.0 java
```

and voila! new JVM. The drawback here is you need to know what the current one is, but that's simple using java -version or by process of elimination.
--
Joseph A. Millar

- **Question:** I would like to know if it could be considered as dangerous to change directly the content of a bytecode.

  Perhaps you'll think the question is strange... In my case, I'm just trying to replace the content of a known string by another string of the same length.

  I have already tried and it seems to work properly, but I'm not sure that it could be OK with different (all) JVM or for example, if the content of the bytecode could be controlled by a checksum or something.

  On JVM there's nothing about these kind of control.

  Answer: Depends on your definition of "dangerous." If you mean "error-prone", then I would answer with a definite yes, especially if you are modifying the code itself... you need to be aware of all the jump targets so you can update them as necessary, and understand enough about the verifier that you can write provably safe code; otherwise a verifier will reject the class file.

  However, a properly written resulting class file is perfectly valid and portable; that's the meaning of Java's much-hyped "binary compatibility".

  There are also no checksums to worry about.
  --
  Chris Smith

- **Question:** what is the difference between "C:\\" and "C:\\." ?
  In the following codes. can anyone explain this ?

  ```
  File dir = new File("C:\\" );
  String [ ] files = dir .list();
  File dir = new File("C:\\." );
  String[] files = dir.list();
  ```

  Answer: "." is used to refer to the current directory. For example, using the change directory command "cd ." changes you to the current directory, effectively doing

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

nothing. "c:\\.\Files\image.jpg" is exactly the same as saying

"c:\\Files\image.jpg"

The code you gave should do exactly the same thing in both forms, to my mind - return a list of the files in the root of the c:\ partition.
--
Lloyd Colling
http://members.xoom.com/lcolling/

- **Question:** This is likely a very silly question. I need to create a .cab file, but I have no idea how to do it...

  Answer: Microsoft has a tool for it. See

  http://msdn.microsoft.com/workshop/management/cab/cab.asp
  You can also get a shareware version of a Cabinet Manager (easier to use than the MS tool) from:

  http://www.microlognet.com/
  --
  Jos

  or you can download a free *.cab tool at:

  http://home.t-online.de/home/lars.hederer/english.htm

- **Question:** Why we can reengineer Java byte code (.class file) back to Java source code? But why binary .exe file we are unable to do it? What is the significant difference?

  Answer: AFAIK, Java byte code goes back to _some_ source, not to the original source. So, reverse engineering is limited.

  > But why binary .exe file we are unable to do it? What is the significant difference?

  (a) There is more than one way to do something using C++.
  (b) There are highly optimizing C++ compilers.
  (c) You won't get the original source anyway. See (a).

  Imagine that your C++ code contains inline functions. The compiler is free do place the body of it replacing a call, or instantiate the function and provide a real call to it. First, there is no way to know when it is going to do one or the other. Second, with some code inlined, how to decide what _was_ an inline function and what wasn't?

  So, the answer is: the distances between the levels of abstraction between byte code and Java source and between machine code and C++ source are significantly different. The bigger the distance, the less possible it is to recreate the source code.
  --
  Victor Bazarov

  in java bytecode all the original variable names are kept. in an exe file smaller symbols are used.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Some .exe decompilers are good enough to convince a jury that the "original" source code was reconstituted. See the Microsoft v. Stac case for an example.

Java is easier, but C is still possible.

- **Question:** I am attempting to write a program that recursively calls a method in order to go down a directory tree, read the information in each directory concerning their files, then write that information to a text file. Are there any built-in methods or programs in Java 1.3 to do this?

Answer: Here is a recursive example for finding a file.

```
static public String Findfile(String dir, String to_find) {
  int i;
  String results;
  String dir_list[]=(new File(dir)).list();

  for (i = 0; i < dir_list.length; i++) {
    File to_test = new File(dir,dir_list[i]);
    if (to_test.isDirectory()) {
        results = Findfile(to_test.getAbsolutePath(),to_find);
        if (results.length() > 0) return results;
    } else {
        if ((to_test.getName()).equalsIgnoreCase(to_find))
           return to_test.getAbsolutePath();
    }
  }
  return "";
}
```

Regards,...Ron

- **Question:** I use the function inside of my class that has the same name in another class of my package. It seems working that program can be confused by this and it seems working better if I use this function within of class like this:

this.XXX();

How does Java distinguish the same methods in different classes?

Answer: Suppose you're inside a method and you'd like to get the reference to the current object. Since that reference is passed secretly by the compiler, there's no identifier for it. However, for this purpose there's a keyword: this. "This" keyword—which can be used only inside a method—produces the reference to the object the method has been called for. You can treat this reference just like any other object reference. Keep in mind that if you're calling a method of your class from within another method of your class, you don't need to use this; you simply call the method. The current this reference is automatically used for the other method. Thus you can say:

```
class Apricot {
  void pick() { /* ... */ }
  void pit() { pick(); /* ... */ }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Inside pit( ), you could say this.pick( ) but there's no need to. The compiler does it for you automatically. The "this" keyword is used only for those special cases in which you need to explicitly use the reference to the current object. For example, it's often used in return statements when you want to return the reference to the current object.
the source: "Thinking in Java" http://www.javafaq.nu/java/book/Contents.shtml

- **Question:** My friend claim that garbage collectors do not collect int value since they are not created with new() method..

  Is he right?

  Answer: Programmers know about the importance of initialization, but often forget the importance of cleanup. After all, who needs to clean up an int? But with libraries, simply "letting go" of an object once you're done with it is not always safe. Of course, Java has the garbage collector to reclaim the memory of objects that are no longer used. Now consider a very unusual case. Suppose your object allocates "special" memory without using new. The garbage collector knows only how to release memory allocated with new, so it won't know how to release the object's "special" memory. To handle this case, Java provides a method called finalize( ) that you can define for your class. Here's how it's supposed to work. When the garbage collector is ready to release the storage used for your object, it will first call finalize( ), and only on the next garbage-collection pass will it reclaim the object's memory. So if you choose to use finalize( ), it gives you the ability to perform some important cleanup at the time of garbage collection.

  source: http://www.javafaq.nu/java/book/Chapter04.shtml#Heading169

- **Question:** In my program where I use finalizers to help GC to free the memory faster... But it seems that it is difficult to know when it happens. I tried to find it and see that even on the same machine my program runs differently...

  Answer: You are right! Garbage collection happens differently each time because it based not on definite schedule but quite complicate algorithms that take into consideration many factors such as CPU load, number of variables, memory size and so on.

  Even developers of JVMs just guess when some process can start but not exactly. Since we have many JVMs, Java programmers, therefore, should avoid writing code for which program correctness depends upon the timely finalization of objects. For example, if a finalizer of an unreferenced object releases a resource that is needed again later by the program, the resource will not be made available until after the garbage collector has run the object finalizer. If the program needs the resource before the garbage collector has gotten around to finalizing the unreferenced object, the program is out of luck.

- **Question:** Does Garbage Collection hang my program for a while?

  Answer: Well, of course it somehow "hangs" if you run your program on one CPU. Not in terms that it hangs until some GC-ing is over.

  Usually well written GC runs in own thread, alongside to your Java program and does

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

not more time than any other thread.

Your program will not wait until some point is reached in GC but rather wait some amount of time which the GC-ing thread allowed to take.

- **Question:** Could you please tell the advantages and disadvantages of having Garbage Collecting in Java?

Answer:

1. Although the programmer still allocates data structures, they are never explicitly freed. Instead, they are "garbage collected" when no live references to them are detected. This avoids the problem of having a live pointer to a dead object.

So, GC "keeps" eye on amount of memory allocated by program and tries to free memory of unreferenced objects in "good" (when your program does not consume much CPU) time.

You do not need to do free() operation like in C++. GC does it for you. Most of memory leaks happen due to bugs in Java itself rather than bad programming (happens also :-))

In a large application, a good garbage collector is more efficient than malloc/free

2. GC makes heap defragmentation (merges the small pieces of free memory into one big piece) that increases performance on the fly. It is difficult to do such thing easy in most of programs written on C++.

3. Your time! If you have fast enough CPU and good GC you will save a lot of time. Manual tuning of a millions pieces of code like malloc/free will take so much time that increases the cost of project dramatically!

Let say like this if you have slow CPU and small program then C++ with malloc/free is more efficient. If you have big one - rely on GC!

4. Security issue: Java programmers cannot crash the JVM by incorrectly freeing memory.

Main disadvantage is that GC adds overhead that can affect performance. In some real time it is critically important that no GC-ing will run in definite periods of time..

- **Question:** Do I need to call Garbage Collector gc() explicitly? If not why then does exist this method?

Answer: Do not afraid, if you do not call gc() it will run anyway! You can be sure that GC-ing happens anyway...
Why does SUN provide us such method?

I see two reasons at least for having it:

1. Time critical applications. If you know that in some moment it is safe to run GC call it. Probably it will run immediately. Not always.

Anyway it can provide better distribution of GC-ing in time increasing GC activity in

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

"safe" time

2. Test applications. It can help you to be sure that your objects will be collected faster than usually.

13. Mobile Java

- **Question:** Please describe shortly J2ME and what is different comparing to J2SE and J2EE? Part 1

  Answer: J2ME is SUN's Java version for machines with limited amount of memory (RAM 128 KB) and low performance (comparing to desktop versions such as PIII) processors.
  The main difference from J2EE and J2SE is that it has a set of different profiles. J2ME consist of three parts:

  1. Java virtual machines* that fit inside the range of consumer devices
  2. APIs that are specialized for each type of device.
  3. A profile, that is, a specification of the minimum set of APIs useful for a particular kind of consumer device (set-top, screenphone, wireless, car, and digital assistant) and a specification of the Java virtual machine functions required to support those APIs

  Each profile is designed for specific hardware type - mobile phone, PDA, microwave oven and so on. Profile contains minimum libraries that are enough to support the particular device. The program is designed with profile for mobile phone does not contain many classes that must be used by oven oven, for example. Each profile uses JVM that uses some subset of JVM from J2SE and J2EE. Usually the program that you write for J2SE or J2EE will not run on J2ME JVM

- **Question:** Please describe shortly J2ME and what is different comparing to J2SE and J2EE? Part 2

  SUN already released two profiles:
  1. The Foundation Profile is a set of Java APIs which, together with the Connected Device Configuration (CDC), provides a J2ME application runtime environment targeted at next-generation, consumer electronic and embedded devices.
  2. The Mobile Information Device Profile (MIDP) is a set of Java[tm] APIs which, together with the Connected, Limited Device Configuration (CLDC), provides a complete J2ME application runtime environment targeted at mobile information devices, such as cellular phones and two-way pagers.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

To be able to write a program with J2ME you must use profile implementation - configuration. Profiles define only specification.

SUN has two profiles now:

1. Connected Device Configuration (CDC) and C virtual machine
   CDC is a Java Community Process effort that has standardized a portable, full-featured Java[tm] 2 virtual machine building block for next-generation, consumer electronic and embedded devices. CDC runs on top of the C Virtual Machine (CVM) that is provided as part of this release.

2 Connected Limited Device Configuration (CLDC) and K virtual machine
   CLDC Java Community Process effort that has standardized a portable, minimum-footprint Java building block for small, resource-constrained devices. CLDC runs on top of Sun's K Virtual Machine (KVM) that is provided as part of this release.

   Also SUN provides developers with J2ME Wireless Toolkit.

The J2ME Wireless Toolkit is a set of tools that provides developers with the emulation environment, documentation and examples needed to develop CLDC/MIDP compliant applications.
--
John S

- **Question:** I know that the PersonalJava application environment is a Java platform for building network-connectable applications for consumer devices for home, office and mobile use.

  How does the PersonalJava application environment differ from the EmbeddedJava application environment?

  Answer: The PersonalJava application environment is designed for Web-connected consumer devices that are often executing applets from a network. To enable this "general purpose" functionality, a core required set of APIs or libraries must be present on every PersonalJava-enabled device.

  The EmbeddedJava application environment is targeted at dedicated-function embedded devices which may be network-connected or standalone. These dedicated-function devices will be performing a defined set of software tasks which are well-known at product conception. Only the class libraries that are needed to support this set of software tasks are included on the device.

  Because of this, every EmbeddedJava implementation may be different since there is no concept of a core required set of API's. The advantage of this is that the EmbeddedJava implementation often occupies a smaller memory footprint than the PersonalJava application environment.

- **Question:** I am going to develop Java programs for wireless devices and found a number of Java technologies for that:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- PersonalJava
- EmbeddedJava
- Java Embedded Server
- J2ME

Could you guide me what should I chose?

Answer: Actually this list is not fully correct - Java Embedded Server does not let you develop programs fro wireless devices. It was designed for small business servers, printers, copiers and point-of-sale (POS) devices such as vending machines, gas pumps, kiosks and cash registers. It can be run only on Solaris, Microsoft Windows NT 4.0, Linux (JDK 1.2), and VxWorks. This technology is obsolete...

PersonalJava as well goes to archive also, no future in the future .-)

http://java.sun.com/products/personaljava/

EmbeddedJava joins this company as well.

Your choice is J2ME!

Question: I am going to start applications development for wireless devices. Could you guide me what I could use for that?

Answer: The production release of the J2ME Wireless Toolkit 2.0 is now available. This version of the toolkit supports the MIDP 2.0 (JSR-118) specification as well as the Wireless Messaging API (JSR-120), and Mobile Media API (JSR-135) optional packages.

The Java 2 Platform Micro Edition, (J2ME) Wireless Toolkit are sets of tools that provide application developers with the emulation environments, documentation and examples needed to develop Java technology applications targeted at CLDC/MIDP compliant mobile phones and entry level PDAs. Two products are available depending on your needs. Both tools are available at no charge to individual applications developers.

For developing applications for MIDP 1.0 devices, use the J2ME Wireless Toolkit version 1.0.4_01.

For developing applications for MIDP 2.0 devices, use the J2ME Wireless Toolkit version 2.0. The 2.0 version also includes built-in support for the Wireless Messaging and Mobile Media APIs.

Below are some of the hot features in J2ME Wireless Toolkit 2.0. You can get more detailed information about the new toolkit by referring to the accompanying User's Guide and Release Notes.

Support for MIDP 2.0
Support for WMA
Support for MMAPI
Solaris and Linux versions
Additional development features:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Integrated Over The Air emulation
Midlet Signing
WMA Emulation features for SMS and CBS messages
New skins for QWERTY and Media devices
Certificate management
Push Registry emulation
Monitoring for all protocols (HTTP(S), Socket, datagram, Comm, SSL, SMS/CBS)
Compile and Runtime selection of API extensions (WMA, MMAPI)
New demo applications
Support for the ProGuard obfuscator

Note: The J2ME Wireless Toolkit 2.0 still includes all of the advanced development features found in Wireless Toolkit 1.0.4_01 (Obfuscation support, method profiling, memory and network monitoring, device speed emulation).

Ł **Question:** What's New in the J2ME Wireless Toolkit 2.0?

Answer: The J2ME Wireless Toolkit contains everything you need to develop MIDP applications.

New Emulator Skins
The J2ME Wireless Toolkit 2.0 includes an entirely new emulator skin, QwertyDevice. Sporting a whopping 640 x 240 pixel color screen and a full keyboard, QwertyDevice represents a wireless communicator, a level beyond a mobile phone.

The Vanguard of MIDP 2.0
The J2ME Wireless Toolkit 2.0 emulator implements the MIDP 2.0 specification, which includes dozens of exciting features, including:

Secure networking with HTTPS
Support for playing simple tones and sampled audio data
The ability to create custom item components for forms
A game API
Use of int arrays to represent and manipulate images
A permission-based security architecture
A push registry that enables MIDlets to receive incoming network connections

Check Out the Sound System
One of the new features in MIDP 2.0 is the ability to play sampled audio and simple audio tones. The J2ME Wireless Toolkit 2.0 supports the full Mobile Media API 1.0.

Setting MIDlet Permissions

Working with Protection Domains

A protection domain determines whether permission is granted or denied.

Signing MIDlet Suites

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

A Killer Feature: Run via OTA
To test installation behavior, the push registry, and signed MIDlets use new option in the Project menu, Run via OTA.

More details can be found here:

http://wireless.java.sun.com/midp/articles/wtk20/

- **Question:** How can I use SSL in MIDP1.0?

  Answer: MIDP1.0 does not have API to use SSL; although I read somewhere that a few implementations were done.

  If you need reliable SSL API then use MIDP2.0

- **Question:** What should I take into consideration when develop an application for wireless device?

  Answer: It is very important to remember that small wireless devices often operate in low quality networks:

  - high network latency.
  - limited network bandwidth.
  - Intermittent connectivity.

  Operator's pricing schemes can be also one of reasons that limit network usage.

  That's why is important to provide data persistence. Persisting data ensures that a user's work session can be suspended and restored rapidly, thus allowing the user to work productively in the interruptible mode common to these use cases. In addition, it helps with recovery from a temporary connectivity problem, such as a problem caused by a localized lack of wireless coverage. You need to provide data refreshing and synchronization.

  You need to have a good balance between connected and disconnected modes, security. Also you probably should use locally stored user/passwords to be able connect automatically every time when connection restores.

  Also you need clearly indicate to user when your application in online and when in offline mode.

- **Question:** Do you know good design patterns for developing applications for wireless devices?

  Answer: yes I found very descriptive list of a few design patterns in "Supporting Disconnected Operation in Wireless Enterprise Applications. A Java BluePrints for Wireless White Paper»:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

"The Facade structural design pattern hides the complexity of the client-side data model implementation...

The Proxy design pattern can be used to abstract the logic that deals with accessing remote data, such as the client/server communication protocol, and any related optimization, such as caching...

The Model-View-Controller (MVC) pattern separates the data model from the presentation (the view and the controller).This pattern isolates the GUI from the data access, which is done via the network from the server or locally from the device's database...

The Adapter design pattern is particularly suited for abstracting the details of dealing with RMS and exposing a simplified, domain-oriented interface to its callers...

The Chain of Responsibilities design pattern is concerned with communication and responsibilities between objects in a control flow. This pattern decouples communicating objects so that you can implicitly send requests to an implicit receiver object through a chain of intermediate objects...

A Factory Method design pattern defines a generic interface that can be used to create an object. However, a class using this pattern defers to its subclasses the decision as to which particular class to instantiate...

A Session Facade design pattern provides a unified interface to a set of interfaces in a system or subsystem...

A Business Delegate design pattern is useful in distributed applications, where remote component look up and exception handling can be complex. A Business Delegate is an intermediate class that decouples application code from business components used by the application..."

I cited only a beginning of every pattern description. More detailed description can be found in the article here:

http://java.sun.com/blueprints/earlyaccess/wireless/disconnected/disconnected.pdf

Ł **Question:** Could you describe the difference between thin and thick wireless application clients and how they related to standalone wireless applications?

Answer: Main criteria that distinguish them are:

- how much data needs to be stored locally and how much remotely
- how much local (in wireless device) processing is required
- how often and how much network connectivity required

I describe them like this:

Thin wireless application client needs low local data storage (-), low processing capability (-) and much network resources (+)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Standalone application I describe like this (+)(+)(-)

And thick wireless application is something between two above listed application.

It can be described like this (+-)(+-)(+-). By "+-" I mean that is in the middle.

- **Question:** How could I start to develop a mobile application? Part 1

Answer: I believe that many of you will forgive me if I direct you towards Ericsson first. The reason is simple - I work in Ericsson :-)

It is up to you to choose for which platform you should develop. In this tip I focus on Ericsson. Later I am going also give you some info about Nokia and other companies.

To be able to get all free documents and development software you need to register first (for free community members).

The Sony Ericsson mobile phones are based on Sun's Java 2 Micro Edition (J2ME) technology. Like many other mobile phones on the market the Sony Ericsson J2ME enabled phones support the Mobile Information Device Profile version 1.0 (MIDP 1.0). In addition to this we have support for sound using Mobile Media API (JSR-135) and SMS using Wireless Messaging API (JSR-120). In the latest we have MIDP 2.0 and some other new API:s that enable a MIDlet to use more features of the mobile phone, and in we will broadly deploy MIDP 2.0 devices in our portfolio during 2004.

Sony Ericsson is very much committed to Java and is actively working in many working groups in the Java Community Process (JCP) and as a member of the J2ME Executive Committee to ensure interoperability and conformance with other vendors. In an effort called "Java Technology for the Wireless Industry" (JTWI) we work together with the rest of the mobile industry to define an interoperability base for devices coming out later in 2003 and 2004. The first result of this work is the JTWI R1 or also called JSR-185. This specification mandates MIDP 2.0 and JSR-120. JSR-135 is optional.

In the future of J2ME, Sony Ericsson expects to have more Java API:s coming out as well as new releases of JTWI.

In the Java community we have also launched a Java Certification program to make Java application testing simpler for the developers. More information about this can be found at http://www.javaverified.com/

- **Question:** How could I start to develop a mobile application? Part 2

The market for Java applications and solutions is significant and Java is a major contributing growth factor in all application areas. All major mobile phone suppliers launched Java-enabled phones for the mass market in 2002 and 2003.

The real test, of course, will be the end user experience. J2ME provides for richer, more real time and exciting applications. It is strongly anticipated that end users will be attracted to Java applications, whether they be in gaming, infotainment, location based services or any other compelling application area. Operators are clearly aware of this, given that they are currently launching trials of J2ME. It is seen as a good

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

way to get existing GSM subscribers to upgrade to GPRS, for example. This will ensure less subscriber churn and new customers won. It can also be easily added to their existing network, without much cost or time.

Getting started

It is very easy to get started developing Java (J2ME) applications for the Sony Ericsson mobile phones.

Step 1 (get to know our phones)

If you are new to J2ME, we suggest that you start by reading the Sony Ericsson Java White Paper (http://www.sonyericsson.com/developer/getDocument.do?docId=64975) to get an understanding of the technology and its implementation in Sony Ericsson phones.

Experienced developers should go straight to the Sony Ericsson Java Developers' Guidelines (http://www.sonyericsson.com/developer/getDocument.do?docId=64976), which provides a deeper insight into the J2ME support in the Sony Ericsson phones.

Step 2 (get the docs & tools)

Next you need the Sony Ericsson J2ME SDK (http://www.sonyericsson.com/developer/getDocument.do?docId=64969), which supports all Sony Ericsson's java phones (Japanese phones excluded) and integrates easily with leading Integrated Development Environment (IDE) from providers such as Borland, Sun and Metrowerks. The SDK contains everything you need to build and test applications for the Sony Ericsson mobile phones, including the support for on device debugging of applications on the T610 and Z600 series of phones, a feature that is very compelling for developers when they want to step through their code and make sure it does not contain any bugs. The Sony Ericsson T610 and Z600 series of phones are the industry's first phones to support on device debugging.

- **Question:** How could I start to develop a mobile application? Part 3

Several developer guidelines, training materials, special white papers and other docs are also available to help you with your development efforts. Get them here >> http://www.sonyericsson.com/developer/site/global/docstools/java/p_java.jsp

Step 3 (start developing)

Now, you have everything you need to start developing high class MIDlets. Should you need technical assistance during your development, you can use the Sony Ericsson Developer World Forums to poll the community or the Sony Ericsson moderators for assistance, or search our Knowledge Base (http://www.sonyericsson.com/developer/site/global/techsupport/knowledgebase/p_knowledgebase.jsp) for known problems.

A collection of Tips, Tricks and Code is also available, as is technical developer support by email.

Step 4 (certify and deploy)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

And last, once you have completed your application, you have the option to have your application certified. The Java community has launched a Java Certification program to make Java application testing simpler for the developers. More information about this can be found at http://www.javaverified.com/ .

Sony Ericsson can help you to bring your applications to the market. Read more about Business Opportunities (http://www.sonyericsson.com/developer/site/global/gotomarket/businessopp/p_businessopp.jsp) or Application Shop (http://www.sonyericsson.com/applicationshop).

The material is taken from:

http://www.sonyericsson.com/developer/site/global/home/techintro/java/p_java.jsp

- **Question:** Does CLDC 1.0 (Connected Limited Device Configuration) support floating point operations?

  Answer: No. But if you use version 1.1 and later (in future) you can use operations with floating point.

- **Question:** Can I use my own Class Loader in MIDP (Mobile Information Device Profile)?

  Answer: No, since it requires significant processing power, and for this and security reasons, no class loader has been implemented in MIDP. Probably in future when we will have more processing power and memory it will be added.

- **Question:** Can I use finalize in MIDP?

  Answer: It does not exist, if an object needs to be cleaned up before it's deleted, the developer must manage that process.

- **Question:** Why can not I use JNI in mobile Java?

  Answer: The reason is simple: On the market exist enormous variety of different mobiles and they become obsolete quite fast. All of them have different HW profiles, different memory and so on. It is just impossible to make any executable code that can run on more than one model....

- **Question:** Can I use RMI in mobiles?

  Answer: No, it is not possible and probably will never be possible. Serialization is not implemented as well.

- **Question:** Can I use thread groups in MIDP

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: No, no thread groups, no daemon threads. As a workaround use collection to store some threads and in such way they will be "grouped". Of course, you can still use multithreading!

- **Question:** If there is no interface for reading or writing to persistent files on the device's file system in MIDP how do I store my data?

  Answer: The Mobile Information Device Profile (MIDP) defines a set of classes for storing and retrieving data. These classes are called the Record Management System (RMS). With the RMS, it's possible to make data persist across invocations of a MIDlet (an MIDP application). Different MIDlets in the same MIDlet suite can also use the RMS to share data. (A MIDlet suite is a set of MIDlets packaged together into a single JAR file.)

  The basic storage mechanism used by the RMS is referred to as a record store. A record store is a collection of records, and a record is basically a byte array of arbitrary data. The size of the byte array can vary for each record. The RMS doesn't understand the contents of a record and doesn't place any restrictions on what goes into the record. The RMS does do some rudimentary bookkeeping, however, such as assigning each record a unique identifier that is valid for the lifetime of the record store.

  A record store is represented by a javax.microedition.rms.RecordStore object.

- **Question:** Does MIDP 2.0 Media API support video and graphics?

  Answer: It depends also on device mobile. Some mobiles support only sound, some also video playback and image capture. Check always manufactures info.

- **Question:** MIDP2.0 supports HTTP, HTTPS, UDP, straight TCP and serial port connections. Will my code for connections look like in J2SE programs?

  Answer: No. You just use a Generic Connection Framework which will abstract all kinds of connections into generic one Connector.

  The code will look like this:

  Connector.open(https://MyBank.com/myaccount.html) or

  Connector.open(socket://abcd.com:10000/)

- **Question:** Do all devices in MIDP2.0 support Push technology?

  Answer: Although MIDP2.0 supports Push technology - waking a MIDlet when request comes from a registered source, device manufactures not always support it in mobiles. You need to check particular model if Push is supported.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I use Timer class in MIDP2.0 to schedule some events in my program and see that some events never start...

  Answer: The possible problem is that some of your events take long time and overlapping with another one. Since Timer is run by single thread, it keeps busy when it must start next event.

  My recommendation is to keep timer events very short and use another threads to continue your task if you think that it will take more time than anticipated.

- **Question:** Does Record Management System (RMS) keep all my data in one record store?

  Answer: No, you allowed creating multiple record stores giving them different names. Record stores are persistent storages and do not get deleted even under reboots.

- **Question:** I know that MIDlets have three states. Is it required to implement all three methods to control all the states?

  Answer: It is strictly recommendable, otherwise you can loose control over a MIDlet behavior.

  MIDlet has three states:

  - Active
  - Paused
  - Destroyed

  You need to implement:

  void startApp()
  void pauseApp()
  void destroyApp(boolean unconditional)

- **Question:** I am going to use Wireless API to send SMSs to another mobile phone.

  Answer: The Wireless API enables MIDlet to send and receive SMSs in either text or binary format

  Be aware that those SMS ate not the same as "normal" SMSs - the other party is always another MIDlet, not just mobile phone...

- **Question:** Can I use code obfuscation for MIDlets and if so which code obfuscators you recommend? Could be good if they were cheap or free... Part 1

  Answer: MIDlets are also Java applications and you can use code obfuscation. It can reduce file sizes by 10-30%...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

I know a few free (GPL or GNU LGPL) code obfuscators that support latest versions of Java, but want to say about those that I tried myself:

- RetroGuard
- ProGuard

RetroGuard: The Java class file format exposes an enormous amount of information about your original Java source code. Using one of the many decompilers on the market, Java class files can be accurately converted back into source format. Your code is therefore open to review and reverse-engineering by competitors and to unauthorized modification and redistribution.

Written in Java (JDK 1.1 through 1.4 compliant).

Designed to be integrated seamlessly into your regular build and quality-assurance process.

Obfuscation is driven by a simple script file, with sensible defaults for all obfuscation options.

To ease the initial generation of the RetroGuard script file, a Java based graphical-user-interface has been provided.

RetroGuard has been rigorously tested to ensure the obfuscated classes will perform identically to the original.

The RetroGuard bytecode obfuscator modifies the contents of your class files in such a way that their behavior is unaffected, while minimizing the amount of information from the source code that is exposed. A side effect of obfuscation is that the class size is reduced, leading to faster download times for your Java applets and applications.

More about RetroGuard read here http://www.retrologic.com/retroguard-main.html

- **Question:** Can I use code obfuscation for MIDlets and if so which code obfuscators you recommend? Could be good if they were cheap or free... Part 2

ProGuard has additional functionality to shrink code size. It can detect and remove unused classes, fields, methods, and attributes. It can then rename the remaining classes, fields, and methods using short meaningless names. The resulting jars are smaller and harder to reverse-engineer.

ProGuard's main advantage compared to other Java obfuscators is probably its compact template-based configuration. A few intuitive command line options or a simple configuration file are usually sufficient. For instance, the following configuration option preserves all applets in a jar:

```
-keep public class * extends java.applet.Applet
```

The user manual explains all available options and shows more examples of this powerful configuration style.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

ProGuard is fast. It only takes a few seconds to process programs and libraries of several megabytes. The results section presents actual figures for a number of applications.

ProGuard is a command-line tool with an optional graphical user interface. It also comes with plugins for Ant and for the J2ME Wireless Toolkit.

More about ProGuard read here: http://proguard.sourceforge.net/

14. Networking, JSP, Mail... etc.

- **Question:** Why do I need JSP technology if I already have servlets?

  Answer: JSP pages are compiled into servlets, so theoretically you could write servlets to support your web-based applications. However, JSP technology was designed to simplify the process of creating pages by separating web presentation from web content. In many applications, the response sent to the client is a combination of template data and dynamically-generated data. In this situation, it is much easier to work with JSP pages than to do everything with servlets.
  --
  This tip is based on SUN's FAQ's

- **Question:** ASP vs. JSP?

  Answer: Comparing JSP with ASP

  Although the features offered by JSP may seem similar to that offered by Microsoft's Active Server Pages (ASP), they are fundamentally different technologies, as shown by the following table:

	JavaServer Pages	Active Server Pages
Web Server Support	Most popular web servers including Apache, Netscape, and Microsoft IIS can be easily enabled with JSP.	Native support only within Microsoft IIS or Personal Web Server. Support for select servers using third-party products.
Platform Support	Platform independent. Runs on all Java-enabled platforms.	Is fully supported under Windows. Deployment on other platforms is cumbersome due to reliance on the Win32-based component model.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Component Model	Relies on reusable, cross-platform components like JavaBeans, Enterprise JavaBeans, and custom tag libraries.	Uses the Win32-based COM component model.
Scripting	Can use the Java programming language or JavaScript.	Supports VBScript and JScript for scripting.
Security	Works with the Java security model.	Can work with the Windows NT security architecture.
Database Access	Uses JDBC for data access.	Uses Active Data Objects for data access.
Customizable Tags	JSP is extensible with custom tag libraries.	Cannot use custom tag libraries and is not extensible.

Tip based on SUN's documents

- **Question:** Can I use HttpURLConnection class to make HTTPS requests in java 1.2?

  Answer: No, you need to use JSSE which includes HTTPS support - the ability to access data such as HTML pages using HTTPS.
  Since Java 1.4 it is the part of J2SE. Just take HttpsURLConnection class.

- **Question:** How can I create a new Instance of InetAddress class? I have not found any constructor...

  Answer: Like this:

```
try { java.net.InetAddress addr =
   java.net.InetAddress.getByName("javafaq.nu"); System.out.println(addr);

}
catch (Exception exc) {
   // do smth here!!!
}
```

Question: What is difference between GET and POST methods?

Answer: There are basically two different methods of getting the data into servlet, which depend on how they were submitted. There are two submission methods, GET and POST, which can both be used by forms. The difference between the two is that using GET, the variables and data will be shown in the page address, but using POST it is invisible. The benefit of GET, though is that you can submit information to the script without a form, by simply editing the URL.

- **Question:** Do you know any P2P (peer to peer) projects like Napster, Morpheus based on Java?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: Interest in peer-to-peer (P2P) technology is growing rapidly, as evidenced by the popularity of services like Napster, Gnutella, and SETI@home. Seeing the potential of P2P computing to enable access to a broader, deeper Web, Sun started a small research effort called Project JXTA.

Until recently, however, P2P technologies have been used primarily in single function applications, such as instant messaging. Taking the concept of P2P much farther, Sun Microsystems founder and chief scientist Dr. Bill Joy conceived the idea of Project JXTA as a means of integrating P2P into the very core of the network architecture.

Project JXTA is a set of simple, open peer-to-peer protocols that enable any device on the network to communicate, collaborate, and share resources. JXTA peers create a virtual, ad hoc network on top of existing networks, hiding their underlying complexity [see Figure 1]. In the JXTA virtual network, any peer can interact with other peers, regardless of location, type of device, or operating environment - even when some peers and resources are located behind firewalls or are on different network transports. Thus, access to the resources of the network is not limited by platform incompatibilities or the constraints of a hierarchical client-server architecture.

Project JXTA technology espouses the core technology objectives of ubiquity, platform independence, interoperability, and security. JXTA technology runs on any device, including cell phones, PDAs, two-way pagers, electronic sensors, desktop computers, and servers. Based on proven technologies and standards such as HTTP, TCP/IP and XML, Project JXTA is not dependent on any particular programming language, networking platform, or system platform and can work with any combination of these.

http://www.jxta.org/project/www/white_papers.html

Question: In FileChannel class I could not find any methods for opening existing files or for creating new ones. So, how I supposed to use the class which I can not "attach to the file"?

It is like having a pump in vacuum: exist a pump but no air :-)

Answer: This class does not define methods for opening existing files or for creating new ones; such methods may be added in a future release. In current releases (1.4.1, 1.4.2) a file channel can be obtained from an existing FileInputStream, FileOutputStream, or RandomAccessFile object by invoking that object's getChannel method, which returns a file channel that is connected to the same underlying file.

- **Question:** I want to use progress bar for downloading files from Internet. But how do I know in advance the file size and then show the progress of downloading?

Answer: First, try to get file size by:

```
int filesize = yourURL.openConnection().getContentLength();
```
if it returns -1 then you really do not know the file size.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Second step in this case will be to use progress bar in indeterminate mode - Mozilla like. This is available since Java 1.4. In this mode, the progress bar displays animation to indicate that work is occurring. As soon as the program determines the length of the task, you should switch the progress bar back into its default, determinate mode.

- **Question:** When I use InetAddress.getByName if a lookup to the name service failed then all subsequent lookups of that hostname fail as well for the lifetime of the virtual machine...
It looks like negative lookups are cashed. What is solution? Is it more bugs I should be aware of?

  Answer: Use latest J2SE 1.4! Prior to J2SE 1.4 negative lookups were always cached. This has changed in 1.4 so that by default if a lookup fails it's possible for a subsequent lookup to succeed. This is particularly important in dial-up environments or environments where the name service is not reliable.

  Also prior to J2SE 1.4.1 the java.net.InetAddress.getLocalHost() would cache the lookup of the IP address of the local machine for the entire session of the application. This behavior poses a problem for applications running on machines that are connected to the internet via a dial-up connection or machines that acquired their addresses through DHCP. This has been corrected in 1.4.1 so that the local IP address is no longer cached. InetAddress.getLocalHost() will return the correct IP address for dial-up connections as well as for DHCP

  Prior to J2SE 1.4 the InetAddress.toString() would perform a lookup via the name service if the hostname wasn't previously resolved. This is no longer the case in 1.4 and toString will return immediately. If the hostname is not resolved the returned string will contain the dotted IP address only.
Applications that depend on toString() returning a string with the resolved hostname should be aware of this correction in 1.4.

- **Question:** Today everywhere I hear "web services, web services". Are not they already in use? Can't I call the services that I use often on the Web like chat, forums, streaming radio and so on - web services? Part 1

  Answer: Yes, indeed quite confusing name... The new technology boom called "web services" came latest few years. Everybody heard and just a few know exactly...
The main difference is that in the web we use client-server model and in the web services - program-to-program model.
I tried to find good description for that "phenomena" and finally found one (written by Maydene Fisher and Eric Jendrock)...

  "Web services, in the general meaning of the term, are services offered by one application to other applications via the World Wide Web. Clients of these services can aggregate them to form an end-user application, enable business transactions, or create new Web services.

  In a typical Web services scenario, a business application sends a request to a service at a given URL using the SOAP protocol over HTTP. The service receives the request, processes it, and returns a response. An often-cited example of a Web service is that of a stock quote service, in which the request asks for the current

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

price of a specified stock, and the response, gives the stock price. This is one of the simplest forms of a Web service in that the request is filled almost immediately, with the request and response being parts of the same method call.

- **Question:** Today everywhere I hear "web services, web services". Are not they already in use? Can't I call the services that I use often on the Web like chat, forums, streaming radio and so on - web services? Part 2

Another example could be a service that maps out an efficient route for the delivery of goods. In this case, a business sends a request containing the delivery destinations, which the service processes to determine the most cost-effective delivery route. The time it takes to return the response depends on the complexity of the routing, so the response will probably be sent as an operation that is separate from the request.

Web services and consumers of Web services are typically businesses, making Web services predominantly business-to-business (B-to-B) transactions. An enterprise can be the provider of Web services and also the consumer of other Web services. For example, a wholesale distributor of spices could be in the consumer role when it uses a Web service to check on the availability of vanilla beans and in the provider role when it supplies prospective customers with different vendors' prices for vanilla beans. "

The tutorial and good deep in description about web services please read here:
http://java.sun.com/webservices/docs/1.2/tutorial/doc/index.html

- **Question:** Is JSP standalone technology or it is an extension of Servlets?

Answer: A JSP page services requests as a servlet. Thus, the life cycle and many of the capabilities of JSP pages (in particular the dynamic aspects) are determined by Java Servlet technology.
When a request is mapped to a JSP page, the Web container first checks whether the JSP page's servlet is older than the JSP page. If the servlet is older, the Web container translates the JSP page into a servlet class and compiles the class. During development, one of the advantages of JSP pages over servlets is that the build process is performed automatically.

This tip based on:
http://java.sun.com/webservices/docs/1.2/tutorial/doc/JSPIntro4.html#wp65950

- Good to read: "Using Web Services Effectively" - the online tutorial begins with a discussion of the Web service processing and interaction models. It describes the two principal Web service models -- a document-oriented model and a Remote Procedure Call-oriented model -- from which the synchronous and asynchronous Web service architectures are derived. Some of the issues to be considered for designing a Web service interface, including granularity of services, the Java API for XML-based Remote Procedure Call (JAX-RPC), and object-centric and document-centric parameter binding.

After covering design issues, the document gives some guidelines on how to architect a Web service. It covers receiving and responding to requests, delegating to business logic, and exposing a Web service. To help illustrate, the document

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

shows how the Java Pet Store sample application has implemented Web services using the J2EE technologies. Whenever possible, recommended practices are highlighted.
http://java.sun.com/blueprints/webservices/using/webservbp.html

- **Question:** What are JSP Model 1 and Model 2 architectures? Part 1

Answer: The JSP specification presents two approaches for building web applications using JSP pages: JSP Model 1 and Model 2 architectures. These two models differ in the location where the processing takes place. In Model 1 architecture, the JSP page is responsible for processing requests and sending back replies to clients.

The Model 2 architecture, integrates the use of both servlets and JSP pages. In this mode, JSP pages are used for the presentation layer, and servlets for processing tasks. The servlet acts as a controller responsible for processing requests and creating any beans needed by the JSP page. The controller is also responsible for deciding to which JSP page to forward the request. The JSP page retrieves objects created by the servlet and extracts dynamic content for insertion within a template.

This model promotes the use of the Model View Controller (MVC) architectural style design pattern. Note that several frameworks already exist that implement this useful design pattern, and that truly separate presentation from content. The Apache Struts is a formalized framework for MVC. This framework is best used for complex applications where a single request or form submission can result in substantially different-looking results.

The answer was found in the article "Servlets and JSP Pages Best Practices":
http://developer.java.sun.com/developer/technicalArticles/javaserverpages/servlets_jsp/

Question: What are JSP Model 1 and Model 2 architectures Part 2

If you are going to use Model 2 in your project I advise you to pay attention to Struts - an open source framework for building web applications.

Struts is part of the Apache Jakarta Project, sponsored by the Apache Software Foundation.

The core of the Struts framework is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, ResourceBundles, and Extensible Markup Language (XML), as well as various Jakarta Commons packages. Struts encourages application architectures based on the Model 2 approach, a variation of the classic Model-View-Controller (MVC) design paradigm.

Struts provides its own Controller component and integrates with other technologies to provide the Model and the View. For the Model, Struts can interact with standard data access technologies, like Enterprise Java Beans and JDBC, as well as most any third-party packages, like Hibernate, iBATIS, or Object Relational Bridge. For the View, Struts works well with JavaServer Pages, including JSTL and JSF, as well as Velocity Templates, XSLT, and other presentation systems.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The Struts framework provides the invisible underpinnings every professional web application needs to survive. Struts helps you create an extensible development environment for your application, based on published standards and proven design patterns.

Struts can be taken here: http://jakarta.apache.org/struts/

- **Question:** I just started to write JSP and servlets code. I read general recommendations how to write a Java code to be readable by other programmers (Code Conventions for the Java Programming Language http://java.sun.com/docs/codeconv/).
But a JSP code is quite different from ordinary Java one... Could you guide me?

  Answer: Yes, you are right. As JavaServer Pages (JSP) technology is becoming widely adopted in web-based applications, many JSP programmers and web developers embarking on developing and maintaining these applications face a dilemma like that of many Java programmers: "How do we structure JSP code that is easier to read, write and maintain consistently?"

  In the article "Code Conventions for the JavaServer Pages Technology Version 1.x Language" (http://developer.java.sun.com/developer/technicalArticles/javaserverpages/code_convention/), you can find a set of standard conventions for writing JSP pages (versions 1.1 and 1.2) that should be followed on a typical software project using web components. The article draws on the Code Conventions for the Java Programming Language as a template to identify various important elements that should be addressed in a coding conventions specification (relevant to JSP technology). In particular, it addresses file names and organization, indentation, comments, directives, declarations, scriptlets, expressions, white space, naming conventions, and programming practices.

- **Question:** Which JSP model (Model 1 or Model 2) and when should I use in my web applications?

  Answer: You should use Model 1 only for simple applications and when you want to get something done quickly.
The main problem with Model 1 is that it is hard to maintain and change (luck of flexibility).

  Model 1 architecture is best when the page navigation is simple and fixed, and when a simple directory structure can represent the structure of the pages in the application. Such applications usually embed the page flow information in the links between the pages. The presence of forward in a JSP page implies that logic embedded in the page is making a decision about the next page to display.

  Model 2 is recommended for medium- and large-sized applications or applications that will grow in complexity in the future.
Over time, as the application grows and changes, page flow logic accumulates. The application becomes difficult to maintain because the page flow logic is distributed across multiple pages. The best time to switch from Model 1 to Model 2 is before this maintenance problem arises. This is why it's usually best to choose Model 2 from the outset, basing the application on an existing Web controller framework that best

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

meets application requirements. Model 1 remains a viable option for simple, static applications.

- **Question:** Has Java some APIs or tools to work with SNMP?

  Answer: Yes, it names Java Dynamic Management Kit (JDMK). The Java Dynamic Management Kit provides a set of Java classes and tools for developing management solutions. It is the first compatible implementation of the JMX specification, the Java Management extensions defined through the Java Community Process.
  about JMX read here: http://java.sun.com/products/JavaManagement/
  Java Dynamic Management Kit uses manageable, Java technology-based components which enable you to quickly create smart, lightweight, embedded network management solutions that can be scaled seamlessly to manage entire networked systems.

  Java DMK agents operate autonomously and offer mechanisms for loading new agent services from the network. At the core of next-generation management solutions, these flexible agents can be created, deployed, updated, enhanced, or removed in real time. In addition, these agents offer remote access through an extendable set of protocols that allows them to be integrated in any type of network environment.

  In addition, the Java DMK provides a complete toolkit for the simple network management protocol (SNMP), the most widespread legacy architecture for network and device management. This gives you the advantages of developing both Java dynamic management agents and managers that can interoperate with existing management systems.

  For a technical overview of the Java Dynamic Management Kit, see the product home page: http://www.sun.com/software/java-dynamic

  P.S. Unlike to other Java products from SUN it is not free and cost some money :-)

- **Question:** Where Can I find more info on Java programming for mobiles?

  Answer: At Ericsson site here:
  http://www.ericsson.com/mobilityworld/sub/open/technologies/java/docs.html
  At Nokia site it is here: http://www.nokia.com/nokia/0,,32913,00.html

  There are two reasons why it is better to start from Ericsson:
  - this company supports Java for mobiles longer than Nokia
  - I am working for Ericsson :-)

- **Question:** Does Java support QoS (Quality of Service) for packets that are sent over the Internet?

  Answer: Yes, Java supports although it also depends on network configuration... Paths through the Internet vary widely in the quality of service they provide. Some paths are more reliable than others. Some impose high call setup or per-packet charges, while others do not do usage-based charging. Throughput and delay also vary widely. Often there are tradeoffs: the path that provides the highest throughput

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

may well not be the one that provides the lowest delay or the lowest monetary cost. Therefore, the "optimal" path for a packet to follow through the Internet may depend on the needs of the application and its user.

Because the Internet itself has no direct knowledge of how to optimize the path for a particular application or user, the IP protocol provides a (rather limited) facility for upper layer protocols to convey hints to the Internet Layer about how the tradeoffs should be made for the particular packet. This facility is the "Type of Service" facility, abbreviated as the "TOS facility" here.

Although the TOS facility has been a part of the IP specification since the beginning, it has been little used in the past. However, the Internet host specification now mandates that hosts use the TOS facility. Additionally, routing protocols (including OSPF and Integrated IS-IS) have been developed which can compute routes separately for each type of service. These new routing protocols make it practical for routers to consider the requested type of service when making routing decisions.

So, in Java you can try use the method setTrafficClass(int tc) of Socket class. As the underlying network implementation may ignore this value applications should consider it a hint.
For Internet Protocol v4 the value consists of an octet with precedence and TOS fields as detailed in RFC 1349. The TOS field is bitset created by bitwise-or'ing values such the following:-

IPTOS_LOWCOST (0x02)
IPTOS_RELIABILITY (0x04)
IPTOS_THROUGHPUT (0x08)
IPTOS_LOWDELAY (0x10)
read more in RFC 1349 and Java API

- **Question:** Does JavaMail support for DIGEST-MD5 authentication in the SMTP provider?

  Answer: Yes, since JavaMail 1.3.1 (June 27, 2003)

- **Question:** What are changes for the HTTP in Java 1.4.2?

  Answer:
  NTLM Authentication
  NTLM authentication for http and https is now supported on Windows platforms. See the Networking Properties document for details of system properties that modify the behavior of this feature.

  Support HTTP redirect to relative URLs
  Prior to 1.4.2, it was required the redirect URLs specified in the Location field of the Http response to be absolute URLs. This is specified by RFC 2616. However, several popular browsers, including IE and Netscape, support relative URLs in the Location field for redirect. Thus beginning with 1.4.2, it was made client behaviours consistent with the popular browsers.

- **Question:** I know that for HTTP we use *.htm or *.html extension. Do I need to use *.shtml for HTTPS?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Not necessarily so. HTML page extensions have nothing to do with the kind of HTTP: HTTP or HTTPS
Historically on UNIXs file extension was *.html. On DOS and first versions of Windows files had so called 8.3 names - 8 characters for name and 3 for file extension.
*.shtml extension generally indicates that SSI (Server Side Includes) technology is used. Server Side Includes (SSI) offer the simplest way to add dynamic content to a Web page. When the Web Server receives a request for a page that may contain SSI commands, it parses the page looking for those commands. If it finds any, they are processed by the Apache module that implements SSI (usually mod_include). The result of this processing - replace the SSI code in the HTML document before it is sent to the requesting user.
SSI can work with other file extensions and defined in Apache configuration.
All extensions - *.htm, *.html, *.shtml - work with HTTP and HTTPS

- **Question:** I am not strong in networking... Is it possible to create two ServerSockets listening in two different ports on one IP address?
If so how can I manage them?

- **Answer:** Sure, you can. Just create them in different threads and call accept methods.

- **Question:** I have difficulty understanding what the difference between sockets and ports. They seem to be doing the same thing, receiving/sending
data streams to other computers over network.

  Answer: A port is a software address on a computer on the network--for instance, the
  News server is a piece of software that is normally addressed through port 119, the POP server through port 110, the SMTP server through port 25, and so on. A socket is a communication path to a port. When you want your program to communicate over the network, you have given it a way of addressing the port and this is done by creating a socket and attaching it to the port.
  basically, socket = IP + ports
  Sockets provide access to the port+ip

- **Question:** Our server (Solaris) accepts a socket when got a request.
Why can we only create 256 sockets? Is it possible to increase the limit?

  Answer: In Solaris and in Unix generally a socket is actually a file descriptor.
  In 2.3 in earlier this requires poking the kernel. In Solaris 2.4+, this can be accomplished by adding the following lines to /etc/system:

  ```
  * set hard limit on file descriptors
  set rlim_fd_max = 4096
  * set soft limit on file descriptors
  set rlim_fd_cur = 1024
  ```

  Raising the soft limit past 256 may confuse certain applications, especially BCP applications. Raising the limit past 1024 may confuse applications that use select().

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Select() cannot use more than 1024 file descriptors at this time prior to Solaris 7. In Solaris 2.6, the RPC code was rewritten to use poll(), which does work with many more fds than select(). Prior to 2.6, all RPC servers will likely crash and burn if you increase the fd soft limit past 1024.

Solaris 7 allows up to 65536 fds passed to select; this is the default for 64 bit applications but it requires recompiling with a larger value for FD_SETSIZE for 32 bit apps.

Programs using stdio or even library calls that use stdio may break when they have more than 256 files open as that is the stdio limit. Programs using many file descriptors should try and reserve a number of low numbered file descriptors for use by stdio.

The stdio limit is removed for 64 bit applications in Solaris 7; if you really need more than 256 FILE * in and you can't use Solaris 7+ or need to run in 32 bits, you can use SFIO from AT&T.

- **Question:** Could you give me an example how to generate TCP/IP packets? I need to write my first network program.

  Answer: There is no need to write such program. Java hides this implementation from you. Just use java.net.Socket or java.net.ServerSocket class and use appropriate constructor.
  Easy and fast!
  If you want even faster, use the method setTcpNoDelay(true) :-)
  --
  AP (J.A.)

- **Question:** How do I change the encoding of the text of characters when I transmit them through the sockets?

  OS uses the default encoding 8859_1. I want 8859_8 encoding to use in outstream.

  Answer: Try using OutputStreamWriter and define setting the encoding in the constructor.

  InputStreamReader and OutputStreamWriter allow you to specify whatever encoding you want in their constructors.

  For example to write to a socket using 8859_8 encoding:

  ```
  OutputStream ostr = socket.getOutputStream();

  OutputStreamReader ostrr = new OutputStreamReader( ostr, "8859_8");
  ostrr.write("This text gets encoded using 8859_8");
  ```

- **Question:** In our application we try to send an image over socket connection in binary format. It works with some systems and does not with others... In some pairs

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

like Linux based Client-Server we get images. When Server and Client reside on different system we often getting strange "digital garbage" after transmission...

Answer: Actually it is known problem and best solution is to use Base64 format.

It will slightly increase the amount of traffic (~30%). An image will be sent as a String...

- **Question:** Is it true that java.nio package name (I mean NIO part) means "New Input/Output"? Part 1.

If so, why it was named like this?

Answer: Yes, NIO stands for "New Input/Output". These APIs -- which were added to the J2SETM platform in Version 1.4 – are designed to boost performance of I/O operations in Java applications. The APIs also provide several features for things like regular expression handling and character-set conversion.

The new I/O (NIO) APIs introduced in v 1.4 provide new features and improved performance in the areas of buffer management, scalable network and file I/O, character-set support, and regular-expression matching. The NIO APIs supplement the I/O facilities in the java.io package.

Features

The NIO APIs include the following features:

- The scalable I/O API will make it easier to write production-quality web and application servers that scale well to thousands of open connections and can easily take advantage of multiple processors;

- The fast buffered binary I/O API will make it easier to write high-performance, I/O-intensive programs that manipulate streams or files of binary data;

- The fast buffered character I/O API will simplify the efficient handling of character streams and files; it will also bring regular expressions and a compact notation for formatted output to the Java platform, putting it on a par with other popular platforms such as Perl;

- **Question:** Is it true that java.nio package name (I mean NIO part) means "New Input/Output"? Part 2

- The character-set converter API will give developers direct access the platform's built-in character-set converters and will also provide for the easy "plugging in" of new converters;

- The new set of I/O exceptions will make it easier to write programs that recover from different types of I/O failures in different ways, and to write user interfaces that behave consistently on different platforms when I/O failures occur; and

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- The new filesystem interface will work more consistently across platforms, will make it easier to write programs that gracefully handle the failure of filesystem operations, will provide more efficient access to a larger set of file attributes, will allow developers of sophisticated applications to take advantage of platform-specific features when absolutely necessary, and will allow support for non-native filesystems, such as network filesystems, to be "plugged in" to the platform.

You could ask: Why were not these needs met by older APIs? Yes it was possible, but it took much time to write custom code... The new API makes this kind of work unnecessary!

This tip is based on SUN's documentation taken from http://java.sun.com

- **Question:** My program wants the capability to place the file physically on to the client machine. Can I use FTP? How? Do I need to read protocol description and implement it from scratch?

  Answer: You can use open source NetComponents package. NetComponents TM provides easy access to FTP, NNTP, SMTP, POP3, Telnet, TFTP, Finger, Whois, and other frequently used Internet protocols. Version 1.3 features newly added BSD R command support.

  NetComponents TM provides one-stop shopping for all your IETF protocol Java programming needs. Access FTP servers from your Java applets, send email from your Java applets, or automate access to Telnet accessible resources. You can do it all with NetComponents TM!

  You will find it here:

  http://www.savarese.org/oro/software/NetComponents.html

  Please feel free to examine the online API documentation and to download the latest version.

- **Question:** I'm having problems with the JavaComm (javax.comm) serial / parallel API from Sun ver. 2.0 for the Win32 platform. It works with JDK 1.1.8 or lower, but I cannot get it to work with Java2 - 1.2.2 and 1.3. It refuses to see my serial ports...

  Answer: JavaComm 2.0 works with all Java 2 distributions including 1.3 and 1.3.1. However there are a few rules:

  1. Install JavaComm for each JRE you are using. Often it's difficult to figure out what JRE is in use. It can happen that you use two different versions from command line and form your IDE where you develop your program.

  2. Install three files into the places:

  in32com.dll --> jre/bin
  comm.jar --> jre/lib/ext
  javax.comm.properties --> jre/lib

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

After that it should work!

- **Question:** We were trying to write a mail client using sun's JavaMail. I was wondering if there is a way to set the priority of the message.
Commercial mail clients does this by setting the X-priority:<num> field of the SMTP header (1 means highest and 5 means lowest - I think the RFC allows much more than this). Looking at the documentation I could not find any way. I was wondering if any of you have done anything similar.

  Answer: Look at MimeBodyPart::addHeader(String name, String value);

  You can add any headers allowed by the RFC spec. :-)

- **Question:** ServerSocket or DatagramSocket? What is better to use in my applications?

  Answer: Both of them are good. It depends on task you do.
  DatagramSocket is designed for UDP and ServerSocket for TCP. TCP is more reliable in terms that it ensures delivery, UDP not. But UDP packets are much smaller and have no such big headers like TCP packets have.
  If it is smaller then there is more probability that it arrives faster. UDP used mostly in areas where you do no mind about retransmission. You just do not need this, because information become obsolete. Sound, for example.
  It is better to make small pause or repeat last packet than to play some piece that was send first but received second.
  It is up to you to take decision. If you do not mind that can lose small piece of data - use UDP. Otherwise use TCP.
  --
  AP. (J.A.)

- **Question:** Why Java networking classes has no Ping method? It is number one utility!

  Answer: I have no answer to this question. Probably because it uses some native interfaces. The good news is that solution already exists and it is free.
  You can take free implementation from our site here:
  http://www.javafaq.nu/java/examples/files/pingicmp.zip
  The program is written by programmed by M Isabel Garcia and Oscar Fernandez.

- **Question:** How can I send/receive SMS messages via GSM phone with Java?

  Answer: Use Kvanttisms - Java class library. The library contains classes for encoding/decoding SMS messages in PDU format and communicating with a GSM terminal through a serial link.
  This library is distributed under Apache Software license, so you will not have problems to use in most of cases.
  Take it here:
  http://sourceforge.net/projects/kvanttisms/
  --
  AP. (J.A.)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** Is it possible to ping an email address? I need to validate entries made on an order entry form, and as part of the validation process I would not only like to validate the formatting of the email address, put also if the address to authentic.

  Answer: No, it is not. If it were possible, spammers could use this to detect valid E-mail addresses en masse. A loop could be written that would detect all active AOL E-mail addresses over a period of days.
  You can easily "validate" an E-mail address as to its form, even sometimes ping the domain at the right of the address (but not always), but you cannot verify that the E-mail address is real without actually posting a message to it. This is why so many E-commerce sites do just that.

  In the past there were some UNIX network services that listed all valid E-mail addresses on a server, but they have largely been discontinued for the same reason -- abuse.
  --
  Paul Lutus, www.arachnoid.com

- **Question:** I wrote small network program using TCP socket. I see that often it does send immediately the data I am trying to send. I have heard that it is because of "nagling".
  What does it mean - nagling?

  Answer: "Nagling" is network transmission algorithm named after John Nagle. This algorithm specified a means of dealing with what he called the small packet problem. This problem arises when we try to send for example one character over the net. Sending of just one byte causes adding additionally ~40 bytes to packet header. So if we send one page of text with small packets it can cause incredible traffic. Overheading will be 4000 % compare to real data.
  John Nagle algorithm avoids such trouble and sends packets when they become big enough.
  In some applications you want to send one byte anyway, for example if you have heart beat that supervise your connection. To disable "nagling" use setTcpNoDelay(true) from java.net.Socket
  --
  AP. (J.A.)

- **Question:** can someone please post a snippet of Java source code that will issue a ping to a specified IP and either return whether successful or not, or round trip time.

  Answer: A solution related to that has been posted a few days ago, an alternative way works for me via Runtime.getRuntime():

```
import java.io.*;

import java.net.*;

class Ping {
  public static void main(String[] args) {
    BufferedReader in = null;
    try {
      Runtime r = Runtime.getRuntime();
      Process p = r.exec("ping 62.2.78.245");
      if (p == null) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
          System.out.println("Could not connect");
        }
        in = new BufferedReader(new InputStreamReader(p.getInputStream()));
        String line;
        while ((line = in.readLine()) != null) {
          System.out.println(line);
        }
        in.close();
      } catch (IOException io) {
        System.err.println(io.toString());
      }
    }
  }
```

--
Linda

- **Question:** I need telnet implemented in Java. I found a few packages on the net. Which is best?

  Answer: It is difficult to say which really best one is until you run for years and get all troubles.
  I like the implementation from Matthias L. Jugel and Marcus Meißner because it supports SSH. It is very important! Ordinary telnet connection exposes everything to any person who uses simplest network tools.
  SSH provides secure connection. This library is free and available under GNU General Public License.
  --
  Leo

- **Question:** I am trying socket level programming through firewalls. Could somebody tell what Http tunnelling is and how to achieve that using Java 2?

  Answer: As an aside, how do you request your proxy server to fetch a page from the net?

  http://developer.java.sun.com/developer/technicalArticles/InnerWorkings/Burrowing/index.html

- **Question:** How can I let dial a phone number with a modem in a Java app.? Is there a way without a System.exec() call and without any M$ classes?

  Answer: You could use javax.comm to do it manually via the serial port and the good old AT command set on the modem. Alternatively, you might look at JTAPI, but that might have its own problems and a lot of overkill.

- **Question:** Does it possible to have two threads running at the same time which reads from the same socket. If a message is received, do both threads then receive it?

  Answer: Two threads can read data from the same socket input stream, but they WON'T each get copies of the same data. They'll each get separate parts of the message.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** how can I get an IP Address in the class InetAddress? The constructor is private, so I can't use it. I want to call the method getName () to get the domain name out of an IP Adress.

  Answer: It is not necessary to construct something :-)
  Just do it like this:
  for example: String hostname = InetAddress.getLocalHost().getHostName();

- **Question:** I'm converting an old java client/server program which is based on raw byte stream heavily into new one which requires utilizing object streams. But if I open input/output object streams on both sides this blocks system and won't proceed...

  Hi,
  I'm converting an old java client/server program which is based on raw byte stream heavily into new one which requires utilizing object streams. But if I open input/output object streams on both side this blocks system and won't proceed.

  ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
  ObjectOutputStream out = new
  ObjectOutputStream(socket.getOutputStream());

  Answer: Upon opening the ObjectInputStream, the constructor blocks to read a header from the stream (doing who-knows-what). Now, what happens is, both your client and server open the InputStream... and happily wait forever for the other side to send them the header they want. Deadlock guaranteed!
  The solution is simple: open the ObjectOutputStream first on at least one side, but better on both sides (usually, symmetry == good). Problem solved
  :)
  You are trying to keep two streams going at once independently, right.
  That means you need at least two threads at each end.

- **Question:** Are there classes available to read and write to an RS 232 port or do this require using native code?

  Answer: See the Java Communications API.

- **Question:** Is there a System property that'll tell me the (TCP/IP) machine name? If not, how do you go about it?

  I need to display all the machine's TCP/IP addresses to the user. This requires a call to InetAddress.getAllByName(), which requires the machine name. You *can't* pass null or "localhost" to this method. Well, you can pass "localhost", but that only gets you the localhost addy itself-
  127.0.0.1. Not useful.

  Answer: Try this: Working that out, I tried

  String hostname = InetAddress.getLocalHost().getHostName();
  System.out.println(hostname);
  InetAddress[] ads = InetAddress.getAllByName(hostname);
  for (int i=0; i<ads.length; i++) {

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    System.out.println(ads[i]);
}
```

by Michiel

- **Question:** Hi, I am new to JTapi (java telephony). I am trying to run a sample code provided with the JTAPI specification. However I get an error message such as :

  Can't get Provider:

  javax.telephony.JtapiPeerUnavailableException: JtapiPeer: DefaultJtapiPeer could not be instantiated.

  Answer:
  As with many Java packages (most notably JDBC) java.telephony is only an API - there is nothing behind it.

  You have to pay someone for an implementation.

  The point is that if you use the API then it will run with anyone's implementation.

- **Question:** Is there any way to connect through the proxy?

  I'm connected to the web behind a proxy. When i try to connect to servers outside they time out and give an exception.
  Is there any way to connect through the proxy?

  Answer: Try this at the command prompt:

  java -Dhttp.proxySet=true -Dhttp.proxyHost=<ProxyHostNameHere>
  -Dhttp.proxyPort=<ProxyPortNameHere> <myclass>

  <ProxyHostNameHere> is to be replaced by the proxy host.
  <ProxyPortNameHere> is to be replaced by a proxy port number (i.e. 8080)
  <myclass> is the java class that you are running.

  You can also set these function through the System class. Check the API for setting properties.
  Hope that works.
  --
  Dan

- **Question:** Socket - My problem is the main class that spawned these threads cannot kill the thread if it's blocked in a read.

  I'm making a program that launches some threads that connect to the net and do some stuff. I'm using a blocking read (TCP/IP BufferedReader.readLine) because I've been recommended by a few people that it was the better way to do it.
  My problem is the main class that spawned these threads cannot kill the thread if it's blocked in a read.
  Is there a thread function I could use? I tried use BufferedReader.ready() to make my read not blocking
  (BufferedReader.ready() returns true only if the the stream is readable otherwise loop so there is no actual blocking) but my problem with that was that

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

BufferedReader.ready() doesn't throw an exception if the other side disconnects so I'm left looping infinitely. I'm really stuck here so if anyone can give me a strategy (I don't need it spelled out to me just general "I used this function type help) I'd really appreciate it.

Answer 1: I've found the only way to unblock a (Socket) read-blocking thread is to close() the socket from another thread. That'll throw an IOException in the read() call.
--
Michiel

Answer 2: Another method that seems to work is to set a socket timeout to something moderately short (like 10 seconds). The blocked read will throw an exception when the timeout occurs. Then the thread can inspect the state of a variable (set by another thread calling an appropriate method) to see if it should terminate.
This approach also makes it trivial to implement a longer inactivity timeout if desired and gives the network thread an opportunity to do some maintenance work if it wants.
--
Peter

- **Question:** I get the impression that only read operations from a Sockets InputStream throw an Exception if the opposite Socket has closed the socket. Writing to the Socket's outputStream works fine...
Is there a way to detect if what I write into a Socket's outputstream is actually still being received by the other side?
Or will I have to check that I actually get a response via the InputStream, and if not, try to open a new Socket and resend the request again?

  Answer: You can try calling flush() after writing the output, but there's no guarantee that you'll get an immediate exception. The underlying TCP/IP software may go through timeout and retry logic before giving up.
  That's probably why you'll usually find out about a broken socket when waiting to read. You should rewrite client so that it sends a request and waits for a response. If it doesn't get a response, it should try to make a new connection and do the request/response thing again.
  If it doesn't work the second time, it should give up for the time being. Perhaps it really makes sense that one has to verify 'by hand' that requests go through.
  --
  Duane Morse

- **Question:** ... I imagine, I'll have to cut my file into small datagrams.

  I just made a little chat program (client and server) and I would like to add the possibility to transfer files.
  The client and the server are communicating with TCP/IP using a socket.
  I imagine, I'll have to cut my file into small datagrams. Is it necessary ? Has someone an idea or a link to source code of such a function ??

  Answer: No. If you were using UDP/IP (DatagramSocket and friends) then it would need to be split up. The point of TCP is to avoid this need. Simple send the entire file in a stream.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I would like to send a large amount of data through a Java socket. I currently use the following (pseudocode):

```
out = BufferedOutputStream(socketOS, 8092)
loop{
  data = byte[100000]
  out.write(data)
}
```

My network connection can send 1 MB in about 5 seconds via FTP, etc., but the above process takes over a minute to send 1 MB.
Is there a way to even come close to my connection's capabilities?

Answer: Once of the main problems with this buffered stream, is that its write method is synchronized. As we all know, synchronized methods can take up to 6 times longer than normal methods to execute. For this reason might I suggest taking the source for BufferedOutputStream, taking out the synchronization, and recompiling as a new class? I would guess this would increase time considerably. Give it a try!
More: one of the unoptimized points is the memory-allocation...
data =byte[100000]... in the loop.
This means, each transferred byte is allocated in memory. That's very bad (very slow)!!!!
--
Chris Shorrock http://www.tantalus.com

- **Question:** Tomcat and packages...

I'm a bit confused about how packages should be organized in Java.
If I work for mycorp, should my classes go into a package named com.mycorp.myapp?
If I do, it seems I must recreate this directory structure in Tomcat in order to use the classes: webapps/myapp/WEB-INF/classes/com/mycorp/myapp ?

Answer: Correct. When you state e.g. package org.foo.cool; in a java source file the compiled file must reside in a directory structure that mirrors this:
org/foo/cool (your source files should/must also reside in a similar directory structure)

The CLASSPATH must be set to the top directory of the packages.
To start an application whose main method is in the org.foo.cool.CoolApp class place the sources in e.g.
/home/foo/javaapps/org/foo/cool/ and start the app with
java -classpath /home/foo/java org.foo.cool.CoolApp
(actually java -classpath /home/foo/java org/foo/cool/CoolApp also works)
--
Nils O. Selasdal

- **Question:** Where can get the more JSP source?

Answer: try

http://www.jspin.com/

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

http://jsptags.com/

http://www.jsptut.com/Getfamiliar.html

- **Question:** our project is currently serializing a lot of stuff. The aim was/is to be able to start up quicker when you (only?) had to de-serialize, instead of initializing all those objects.

  My **Question:** is this really a start up time winner or the opposite?

  Answer: My experience with serialization is limited, precisely because it took so much time to deserialize - and so I abandoned using it.

  Well, the only way to tell for sure is to try it both ways and see which one is faster.

  If I had to guess, I'd say object initialization is faster. But I don't know what you code does to initialize its objects.
  Does it just set some fields to default values? Or does it set its fields to values returned from a website accessed over a 9600 baud modem? It makes a difference.

  If you want your program to start up fast, the way to do that is:

  1) Write your program in the most straightforward way, not thinking about performance until it's done and feature complete.

  2) Check to see if it's fast enough. If so, stop.

  3) If not, use a performance-measuring tool to see where it's spending its start up time; optimize those routines.

  Any other methodology is invalid, in that it is as likely to slow your app as speed it. (I'd guess that's what you did.)
  --
  Marshall Spight

- **Question:** I've tried in every possible way I could find to determine my actual internet IP address with Java. It just doesn't work. My computer is connected through a router that connects to my ISP. This causes the InetAddress.get* methods to return only an intranet ip address.

  InetAddress.getAllByName("<router ip address>") doesn't work either, it returns the intranet address of the router.

  Answer: If you can't see "your" IP address using getAllByName(), then it is not "your" IP.

  If the following code cannot resolve your Internet IP address, then you don't have a direct connection to the Internet:

  ```
  // NetTest.java

  import java.net.*;
  ```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
class NetTest {

   public static void main(String args[]){
      try {
         String hostName = InetAddress.getLocalHost().getHostName();
         InetAddress[] ipList = InetAddress.getAllByName(hostName);
         System.out.println(hostName + ":");
         for(int i = 0;i < ipList.length;i++) {
            System.out.println(ipList[i].getHostAddress());
         }
      }
      catch(Exception e) {
         e.printStackTrace();
      }
   }
}
```

My example run:

```
pl-alpha:
192.168.0.2
228.232.218.126
```

The first is my fixed intranet address; the second is my temporary log-on address.

If the NetTest (getAllByName) is returning only private IPs for you (192.168.*, 10.*) it sounds like the router is providing NAT and Masquerading so that the PC doesn't really have a real IP address--it's like you're behind a firewall. There's no way for clients on the internet to reach your server.

On the other hand, if you can configure your router to forward packets on a particular port to your intranet address, then you can just bind to the local address and the rest will come out ok.
--

Paul Lutus
www.arachnoid.com

15. Operation Systems and Java

-

- **Question:** Is anyone aware of any way to determine OS type, version...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: There is in the java.lang.System
try first:

```
Properties p = System.getProperties();
System.out.println(p);
```

to see what properties you can get(CPU among others) you then can:

```
String prop = System.getProperty("a property(the CPU among other)");
```

- **Question:** Does anyone know if there is an easy way to invoke UNIX shell scripts from a Java application.

Answer: There is!
Execute a shell and then write the name of the shell script on the shells stdin stream.

```
Process child = rtime.exec("/bin/bash");
BufferedWriter outWriter = new BufferedWriter(new
OutputStreamWriter(child.getOutputStream()));
outWriter.writeLine("filename");
outWriter.flush();
```

Question: I am using MouseWheelListener in my program and it works fine on Windows, but not on Linux...

Answer: To use a mouse wheel in Linux you must enable it in OS first.
Two modifications to the /etc/X11/XF86Config file are required. Under the "Pointer" section:

Add the line:

```
ZAxisMapping 4 5
```

Change the protocol to: "imps/2" (this will vary depending on your particular wheel mouse)

- **Question:** We need to authenticate a user on the local Linux system before he/she/it can log on to the application and begin taking over the world. Since the Linux system uses shadow passwords, we must have a method to retrieve the password and authenticate the user in the user shadow database. Once authenticated, this user can work with the application.
How it can be done in Java?

Answer: It can be done, it is already done!
Here if you are interested in interfacing C with Java in Linux (the JNI Solution):
http://cscene.org/CS4/CS4-04.html
--
John

- **Question:** Is there a way to differentiate the enter key on the main keyboard from the enter key on the keypad?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: I don't think so, they're both represented by the character code 10.

- **Question:** Is there any way to manage Systray Icons with Java? I didn't even find anything in the Microsoft packages...

  Answer: You must create a C/C++ program that call directly the NotifyIcon API to display/manage the Icon in the Systray. Then call the program from Java with JNI.

- **Question:** Currently I'm running two operating systems on one machine, Windows 2000 and Windows ME...

  Currently I'm running two operating systems on one machine, Windows 2000 and Windows ME.
  Windows ME runs on default, Windows 2000 is on D drive and ME is on C drive. I tried to add the JDK directory to the classpath but there isn't a autoexec.bat on the D directory, but there is one for C.
  Should I just create a autoexec.bat for D?

  Answer: Go to Settings/Control Panel/System/Advanced/Environment Variables... and edit your CLASSPATH variable if exists or add a new one.

- **Question:** I need to be able to run a shell script from java, by doing a host call to UNIX. I am currently trying 'Runtime.exec' to do this. It says it runs it, but doesn't actually successfully complete the command (which is a file delete).

  I need to be able to run a shell script from java, by doing a host call to unix. I am currently trying 'Runtime.exec' to do this. It says it runs it, but doesn't actually successfully complete the command (which is a file delete).
  A few questions I have about this:

  1. Can I trace it or something to see why it isn't working?
  2. How can I get the 'return code' from the shell script?
  3. Will the java procedure wait for the shell script to execute, or does it run a separate thread concurrently?

  Answer: > 1. Can I trace it or something to see why it isn't working?

  Runtime.exec() returns a Process object that you can get information from.

  If the script is written with sh or some derivate thereof you can do 'set -x' in the script to trace its behaviour. For other scripting languages there is likely a similar mechanism. To see the trace in your Java program, read from the error or output streams of the process (Process.getErrorStream() or Process.getOutputStream()). It's possible that the script is already printing a message to its error stream that you aren't seeing, indicating why it isn't working.

  Answer: > 2. How can I get the 'return code' from the shell script?

  Call Process.exitValue().

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: > 3. Will the java procedure wait for the shell script to execute, or does it run a separate thread concurrently?

The program runs concurrently in a separate *process*, not a thread within the JVM. To wait for the process to finish, call Process.waitFor().

- **Question:** Are not-initializable classes and methods like System.out.println() and Math.random() "synchronized" ?

Answer: I think they are synchronized. Simple observation: did you ever see once that printout was broken into two pices by another printouts? I mean for example if you do:

In 1st thread: System.out.println("1234567890");
And in 2nd thread: System.out.println("something else here");

it never will be broken like:

12345
something else here
67890

Even if Sun didn't write about it explicitly, we can see that it is synchronized or at least behaves like synchronized that is the same for our real life.

- **Question:** I have a little problem with the JVM (1.3) under Linux (SUSE 7.1). Every time I try to run a java application that includes a GUI, while the JVM starts I get a few warning messages saying that the JVM can't locate some fonts. The application continues to run but non of the fonts included in my code are used, instead I get the same font in every part of the GUI.

Answer: I get the same response. This is because the fonts specified in the code are not resident in the place where the JVM looks for fonts. They may actually be on your Linux machine somewhere - there are hundreds of fonts available to X windows in Linux.

There are only 5 fonts which are guaranteed to be available in every java environment:
Serif, SansSerif, Dialog, DialogInput, and Monspaced.
There are other fonts available to you on your system, of course. But the problem with using any fonts other than the guaranteed five is that you cannot be assured your user will have the fonts you have specified in you code, on his/her system.

- **Question:** I want to know, if I have more than one CPU on my machine, is it possible to bind a JVM to a particular CPU?

Answer: That's operating-system dependent..... Java has no control over this. For example, under Solaris, look at pbind(1M)...

- **Question:** How can I pass a string to the command line(DOS)? Also I want to capture the output given by the command line in a string.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Try this out:

```
// works for DOS
String cmds[] = new String[2];
cmds[0] = "dir"; // replace with "ls" on UNIX
cmds[1] = "c:"; // replace with "/" on UNIX

// execute the command
Process pro = Runtime.getRuntime().exec(cmds);
// wait until it's done executing
pro.waitFor();

// what did the process output from the Input pipe back to
// this process (okay, who named this stuff)?
InputStream out = pro.getInputStream();

// output it (really slowly)
int i;

while ((i = out.read()) != -1) System.out.println((char) i);
```

- **Question:** How can I take a program that runs in a DOS shell and send the text that comes from the shell program into a Java program where it can analyzed, etc.?

Answer: From a command line, use a pipe (with the "|" symbol):

```
c:\> dosprogram | java JavaProgram
```

In the Java program, read the text from System.in:

```
public static void main(String[] args) throws IOException {
    int nLines = 0;
   BufferedReader in =
   new BufferedReader(
   new InputStreamReader( System.in));
   for (;;) {
      String line = in.readLine();
      if (line == null)
         break;
      nLines++;
      System.out.println(nLines + ":" + line);
  }
}
```

- **Question:** If there is a way to run a java program by just typing the name in UNIX. I mean instead of typing for example "java Main" just type "Main" and run the program. And how do I implement that in a makefile?

Answer: Write a script that runs the program and put it in your path. For instance:

```
#!/bin/sh
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

java BlahBlah

Call this whatever you want, mv it to your /usr/local/bin directory, then just type it at the command line and BlahBlah will be run.

- **Question:** Are there any Java libraries for executing Linux commands?

  Answer: Try java.lang.Runtime.exec(). E.g.

  ```
  Runtime.getRuntime().exec("xterm");
  ```

  Note if you want to use shell builtins or shell features like redirection you need a shell, e.g.:

  ```
  Runtime.getRuntime().exec(new String[] {"/bin/sh",
                            "-c", "ls / 2>&1 | tee ls.log"});
  ```

- **Question:** I'd like to know how to know which operating system java application is running on.

  Answer: You could try using the system Properties.
  e.g.

  ```
  Properties prop = System.getProperties();

  String osString = prop.getProperty( "os.name" );
  ```

- **Question:** I would like to know how my Java program can catch when someone sends a "kill" to my app in Unix or does a Ctrl-C in windows?

  In Unix there is atexit() function that handles this type of situation. Is this possible in Java?

  Answer: Starting with 1.3 there is Runtime.addShutdownHook(). This is for cleanup only.
  from API: "A shutdown hook is simply an initialized but unstarted thread. When the virtual machine begins its shutdown sequence it will start all registered shutdown hooks in some unspecified order and let them run concurrently. When all the hooks have finished it will then run all uninvoked finalizers if finalization-on-exit has been enabled.
  Finally, the virtual machine will halt. Note that daemon threads will continue to run during the shutdown sequence, as will non-daemon threads if shutdown was initiated by invoking the exit method. Once the shutdown sequence has begun it can be stopped only by invoking the halt method, which forcibly terminates the virtual machine.
  Once the shutdown sequence has begun it is impossible to register a new shutdown hook or de-register a previously-registered hook. Attempting either of these operations will cause an IllegalStateException to be thrown.
  Shutdown hooks should also finish their work quickly. When a program invokes exit the expectation is that the virtual machine will promptly shut down and exit. When the virtual machine is terminated due to user logoff or system shutdown the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

underlying operating system may only allow a fixed amount of time in which to shut down and exit. It is therefore inadvisable to attempt any user interaction or to perform a long-running computation in a shutdown hook."
More read here: http://java.sun.com/j2se/1.3/docs/api/index.html

- **Question:** Is it possible for a minimized window in MS Windows task bar to start blinking? In any IRC client a minimized window starts blinking when a new message appears in it. Is it possible in Java?

  Answer: I doubt very much that it is possible with pure java; you would have to use some native code to achieve that. Maybe what you could try though, is to play around with the icon used in the title bar of the frame. I haven't tried this myself - and in any case I'm running linux, but it might be possible to have two icons and to switch them around at some rate using the Frame.setIconImage(Image) method. This will only affect the icon itself - not the whole window, but it's better than nothing.
  --
  Allen Wallis

- **Question:** How do I find the list of all system properties?

  Answer: do smth like this:

```
Enumeration list = System.getProperties().propertyNames();
   while(list.hasMoreElements()){
     System.out.println((String) list.nextElement());
   }
}
```

- **Question:** Is it cls-like command in DOS (CMD) window? I would like to clean the screen the user sees?

  Answer: Unfortunately there is no command as "cls", but try this instead:

```
for   (int i=0; i<25; i++)  System.out.println();
```

  It will print 25 empty lines and they will move current printouts up.

- **Question:** Why my program does not give the address of the local machine on one PC and give on another?

  Answer: As long as you have TCP/IP installed, you should at least get 127.0.0.1

- **Question:** Can applet corrupt my registry file?

  I have written a java applet for displaying my genealogy. It has worked fine at home. My brother tested it on my web site with no problems. BUT, the third person I asked to take a look at it reported, that though it worked fine, when he exited the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

page, his system locked up and he had to reboot. After the reboot he said his registry was corrupted and had to be restored.

Has anyone seen anything like this? if so, do you have any suggestions as to probable cause and fix?

Answer: Any time you don't shutdown properly, your registry can be corrupted, even if the program that did the freezing was not even using the registry.

The registry is one of the stupidest ideas ever conceived in computer science. It puts ALL your eggs in one basket.
--
by Roedy Green
For the JAVA GLOSSARY see http://mindprod.com/jgloss.html

- **Question:** Does anybody know how to find out the complete path of a system default browser (if there's such) from a Java stand alone app?

Answer: Under Windows you can 'invoke' the default browser calling 'start filename.html', under Unix is a lot more complicated because nobody can assure you that a browser exists and the user is under X....
--
Davide

- **Question:** ...The problem is that, after the file is uploaded to the server, every end of line is placed with ^ M character...

I have written a servlet which uploads a file from client(Win32) to server (Unix). I have used DataInputStream to read the file in bytes and FileOutputStream to write the file to the location in the server.

The problem is that, after the file is uploaded to the server, every end of line is placed with ^ M character. This problem is faced only when I upload to Unix Server from Windows O/S.

Answer: This is normal, I think. A line ends with a single return symbol in Unix and ends with a return symbol and a switching line symbol in Windows. The editor in Unix shows "^ M" when it meets the switching line symbol.
So to fix this bug, you can remove the switching line symbol in the end of lines when you upload files from Windows to Unix.
Or Use a BufferedReader to read the source file (line by line) and a BufferedWriter to write the destination file. That way the source line separators will be discarded and the correct separators (according to the target OS) will be used.
If you only have an InputStream to start with, use InputStreamReader to convert it to a reader.
--
Jorge

- **Question:** I have a server written in Java that I would like to start up as a NT service...does anyone know how to do this?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: Check this site:
http://www.kcmultimedia.com/smaster/

I read there:
"ServiceInstaller is a FREE utility that makes it easy to install programs as Windows NT Services, including pure Java applications, without manually editing the registry!"

- **Question:** Does any one know how to kill or stop a process through java?

Answer: It must be a process you created, and there are rare occasions when this will not work.

```
Process p = Runtime.getRuntime().exec(path);

// later ...

p.destroy();
```

--
Paul Lutus, www.arachnoid.com

- **Question:** I'd like to determine the free disk space in a platform independent way. So far, I've found no Java way to do so...

The best I've been able to come up with is running the UNIX "df" utility (which is available for a number of non-UNIX platforms, too, such as Win32 and OS/2).

Answer: Samizdat Productions Releases JConfig 2.1.1 JConfig is a class library that extends the core Java API. It lets you work with files, web browsers, processes, file types, and other system-level items in a much more advanced manner than that provided by the standard Java class libraries. A list of JConfig's features is given below.

JConfig is for free most freeware and educational projects, and it now comes with the complete Java and C++ source code!

** Download JConfig here:
http://www.tolstoy.com/samizdat/jconfig.html
Here's a partial list of JConfig's features, by category:
Files:
Enumerate the user's disk drives, and obtain extended information on files, directories, volumes, and filesystems: their icons, creation dates, version information, mount points, and more...
Web Browsers:
Launch a file or URL in the user's Web browser...
Video Monitors:
Enumerate and get information on the user's video monitors: bit depth, bounds, and more...
External Processes:
Create external processes, send basic commands to external processes, obtain the PSN or HWND of a process you created, and enumerate the currently running processes...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

File Types:
Find applications associated with a given file type, find applications by name, and convert between Windows file extensions and Mac creator/file type codes...
--
boruvek

16. RMI

Question: What is difference between RMI and Java Messaging? And between email and Java messaging.

Why do we need it?

Answer: "Messaging is a method of communication between software components or applications. A messaging system is a peer-to-peer facility: A messaging client can send messages to, and receive messages from, any other client. Each client connects to a messaging agent that provides facilities for creating, sending, receiving, and reading messages.

Messaging enables distributed communication that is loosely coupled. A component sends a message to a destination, and the recipient can retrieve the message from the destination. However, the sender and the receiver do not have to be available at the same time in order to communicate. In fact, the sender does not need to know anything about the receiver; nor does the receiver need to know anything about the sender. The sender and the receiver need to know only what message format and what destination to use. In this respect, messaging differs from tightly coupled technologies, such as Remote Method Invocation (RMI), which require an application to know a remote application's methods.

Messaging also differs from electronic mail (e-mail), which is a method of communication between people or between software applications and people. Messaging is used for communication between software applications or software components."

From SUN tutorial:
http://java.sun.com/products/jms/tutorial/1_3_1-fcs/doc/overview.html#1027335

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I work with RMI now and need to implement a new feature into software that was written by another people one year ago. I need to find out: what is IP address of a client and then connect or reject the connection. I do not want to add methods to client... It must be the way how to find out it using RMI methods. Unfortunately I did not find with first attempt.

  Answer: Your RMI server extends UnicastRemoteObject that extends java.rmi.server.RemoteServer.
  That class (RemoteServer) has public static String getClientHost() method.
  When called from a thread actively handling a remote method invocation the hostname of the client is returned.

  Then use public static InetAddress getByName(String host) method of java.net.InetAddress
  Be aware that you do need to try to create an instance of InetAddress. It is static. Just get it!

- **Question:** What do I have to do to get RMI to operate through two firewalls?

  Answer: First of all, what co-operation can you expect from the client-side firewall? In the most pessimistic case, the client-side firewall allows no direct TCP connections and has only a HTTP proxy server so that firewalled clients can "surf the web". In this case, your server host will receive connections at port 80 containing RMI requests embedded in HTTP requests. You can use a HTTP server with the java-rmi.cgi program, or you can run the RMI server directly on port 80. Either way, the server cannot use call back objects exported by the clients.

  A more optimistic case is that the client can make direct connections to the server but cannot receive incoming connections from the server. In this case, call back objects are not normally possible either. (There is a specific case where applets in some environments may trigger a multiplexed connection mode intended for JDK 1.0.2 compatibility -- this would allow call backs but is not scalable or supported.)

  The most conservative approach, assuming no help from the client firewall administrators, is:

  Avoid using callback objects
  Run your servers on `public' ports such as 80, 81, 8001, or 443
  If the servers are not running on port 80, either:
  Put a CGI-capable HTTP server on port 80 using the java-rmi.cgi script; or
  Run a port redirector (such as DeleGate) on port 80, which will accept connections and immediately connect to the real server port to pass bytes back and forth. This will cause getClientHost() to return misleading information, so don't make the Registry available through this method unless it's on a different host.
  source: http://java.sun.com/products/jdk/1.2/docs/guide/rmi/faq.html

- **Question:** Why doesn't my server program exit when all clients disconnect? (RMI)

  Answer: In RMI, a server VM should exit if there are:
  No outstanding client-held references to the remote objects in VM, and
  No non-daemon threads executing in the VM.
  However, just because there are no local or remote references to a remote object does not mean the object will be garbage collected in a timely fashion. It does mean

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

that the remote object's memory can be collected to satisfy a memory allocation that would otherwise fail (with an OutOfMemoryError).
Although the Java API does not specify the timeliness of collection anyway, there is a particular reason for the what can seem like indefinitely-delayed collection of remote objects in the JDK 1.1 implementation. Under the covers, the RMI runtime holds a weak reference to exported remote objects in a table (to keep track of local as well as remote references to the object). The only weak reference mechanism available in the JDK 1.1 VM uses a non-aggressive, caching collection policy (well-suited for a browser), so objects that are only "weakly reachable" will not get collected until the local GC decides that it really needs that memory to satisfy another allocation. For an idle server, this could never happen. But if memory is needed, an unreferenced server object will be collected.

The JDK1.2 release will include a new infrastructure that RMI will use to reduce significantly the number of conditions under which this problem occurs.
source: http://java.sun.com/products/jdk/1.2/docs/guide/rmi/faq.html

- **Answer:** How does the distributed garbage collector detect a client (RMI) that disconnects? Is it advisable to use System.exit for graceful client termination?

Answer: When the RMI runtime in a client VM detects that a remote object is no longer referenced locally, it asynchronously notifies the server relatively quickly so that the server can update the object's referenced set accordingly. The distributed garbage collector uses a lease associated with each client-held remote object reference, and renews leases to remote objects while the client still holds such references. The purpose of the lease renewal mechanism is to allow the server to detect the abnormal termination of clients, so that a server does not hold on to a remote object forever because of a client that was not able to send the appropriate "unreferenced" message before it stopped running. In this context, a client invoking System.exit() is considered abnormal termination, because it does not allow the RMI runtime to send the appropriate "unreferenced" messages to the server. Executing System.runFinalizersOnExit in the client before termination is not sufficient, because not all of the necessary processing is handled in a finalizer; i.e. the "unreferenced" message will not get sent to the server. (Using "runFinalizersOnExit" is generally ill-advised and deadlock-prone anyway.)
If you need to use System.exit() to terminate a client VM, to ensure that remote references held in that VM are cleaned up in a more timely fashion, you should make sure that there are no remote references still reachable (explicitly null any local references to make them unreachable from running threads), and then it helps to do something like this before exiting:

```
System.gc();
System.runFinalization();
```

source: http://java.sun.com/products/jdk/1.2/docs/guide/rmi/faq.html

- **Question:** Why can't I get an immediate notification when a client crashes? (RMI) Part 1

Answer: Because nothing the server does can distinguish a crashed host from a network delay.
If or when the crashed client later restarts and contacts the server, the server can infer that the client has lost its state. If a TCP connection is held open between the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

client and the server throughout their interaction, then the server can detect the client reboot when a later attempt to write to the connection fails (including the hourly TCP keep alive packet, if enabled). However, RMI is designed not to require such permanent connections, as it impairs scalability and doesn't help very much.

Given that it is absolutely impossible to instantly determine when a network peer crashes or becomes otherwise unavailable, you must decide how your application should behave when a peer stops responding.

The major tools you have for this task are timeouts and resets. After a timeout, you may conclude that a peer is unreachable, but the peer must be aware of the timeout so that it gives up trying to reach you. The leasing mechanism is designed to do this semi-automatically.

A reset is a purge of existing state held for a peer. For example, a client may cause a reset when it first registers with its server, causing the server to discard any previous state held for that client (having deduced that the client has restarted without memory of the previous, dead, session).

- **Question:** Why can't I get an immediate notification when a client crashes? (RMI) Part 2

Frequently, the aim is to have and maintain a definitive list of clients at the server, and to keep it up-to-date without error or failure. Since failure and delay can happen at any time in a networked system, some degree of error in the list must be anticipated. If a lease or other mechanism is used to enforce a timeout, then the problem of resource leakage is solved. If the problem of stale data is more serious -- that is, if it would interfere with correct operation -- then it must be explicitly purged in cases where it would otherwise have an effect.

For example, if a business object is locked for editing by a human, and the session dies, then the lock must be broken somehow. In this case, the lock would need a timeout, but if the same human logs in immediately and expects not to have to wait for the timeout to expire, the new session must either take over the lock or assert that the user holds no locks (allowing the server to safely kill the lock).

source: http://java.sun.com/products/jdk/1.2/docs/guide/rmi/faq.html

- **Question:** Why must classes implement Serializable in order to be written to an ObjectOutputStream? Part 1.

Answer: The decision to require that classes implement the java.io.Serializable interface was not made lightly. The design called for a balance between the needs of developers and the needs of the system to be able to provide a predictable and safe mechanism. The most difficult design constraint to satisfy was the safety and security of classes for the Java programming language.
If classes were to be marked as being serializable the design team worried that a developer, either out of forgetfulness, laziness, or ignorance might not declare a class as being Serializable and then makes that class useless for RMI or for purposes of persistence. We worried that the requirement would place on a developer the burden of knowing how a class was to be used by others in the future, an essentially unknowable condition. Indeed, our preliminary design, as reflected in the alpha API, concluded that the default case for a class ought to be that the objects in the class

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

be serializable. We changed our design only after considerations of security and correctness convinced us that the default had to be that an object not be serialized.

- **Question:** Why must classes implement Serializable in order to be written to an ObjectOutputStream? Part 2.

Security restrictions
The first consideration that caused us to change the default behavior of objects had to do with security, and in particular in the privacy of fields declared to be private, package protected, or protected. The Java platform restricts access to such fields for either read or write to a subset of the objects within the runtime.
No such restriction can be made on an object once it has been serialized; the stream of bytes that are the result of object serialization can be read and altered by any object that has access to that stream. This allows any object access to the state of a serialized object, which can violate the privacy guarantees users of the language expect. Further, the bytes in the stream can be altered in arbitrary ways, allows the reconstruction of an object that was never created within the protections of a Java platform. There are cases in which the re-creation of such an object could compromise not only the privacy guarantees expected by users of the Java platform, but the integrity of the platform itself.

These violations cannot be guarded against, since the whole idea of serialization is to allow an object to be converted into a form that can be moved outside of the Java platform (and therefore outside of the privacy and integrity guarantees of that environment) and then be brought back into the environment. Requiring objects to be declared serializable does mean that the class designer must make an active decision to allow the possibility of such a breach in privacy or integrity. A developer who does not know about serialization should not be open to compromise because of this lack of knowledge. In addition, we would hope that the developer who declares a class to be serializable does so after some thought about the possible consequences of that declaration.

- **Question:** Why must classes implement Serializable in order to be written to an ObjectOutputStream?

Note that this sort of security problem is not one that can be dealt with by the mechanism of a security manager. Since serialization is intended to allow the transport of an object from one virtual machine to some other (either over space, as it is used in RMI, or over time, as when the stream is saved to a file), the mechanisms used for security need to be independent of the runtime environment of any particular Java virtual machine. We wanted to avoid as much as possible the problem of being able to serialize an object in one virtual machine and not being able to deserialize that object in some other virtual machine. Since the security manager is part of the runtime environment, using the security manager for serialization would have violated this requirement.

Forcing a conscious decision.
While security concerns were the first reason for considering the design change, a reason that we feel is at least as convincing is that serialization should only be added to a class after some design consideration. It is far too easy to design a class that falls apart under serialization and re-construction. By requiring a class designer to declare support for the serialization interface, we hoped that the designer would also give some thought to the process of serializing that class.
Examples are easy to cite. Many classes deal with information that only makes sense

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

in the context of the runtime in which the particular object exists; examples of such information include file handles, open socket connections, security information, etc. Such data can be dealt with easily by simply declaring the fields as transient, but such a declaration is only necessary if the object is going to be serialized. A novice (or forgetful, or hurried) programmer might neglect to mark fields as transient in much the same way he or she might neglect to mark the class as implementing the Serializable interface. Such a case should not lead to incorrect behavior; the way to avoid this is to not serialize objects not marked as implementing Serializable.

Another example of this sort is the "simple" object that is the root of a graph that spans a large number of objects. Serializing such an object could result in serializing lots of others, since serialization works over an entire graph. Doing something like this should be a conscious decision, not one that happens by default.

- **Question:** Why must classes implement Serializable in order to be written to an ObjectOutputStream? Part 4.

The need for this sort of thought was brought home to us in the group when we were going through the base Java API class libraries marking the system classes as serializable (where appropriate). We had originally thought that this would be a fairly simple process and that most of the system classes could just be marked as implementing Serializable and then use the default implementation with no other changes. What we found was that this was far less often the case than we had suspected. In a large number of the classes, careful thought had to be given to whether or not a field should be marked as transient or whether it made sense to serialize the class at all.

Of course, there is no way to guarantee that a programmer or class designer is actually going to think about these issues when marking a class as serializable. However, by requiring the class to declare itself as implementing the Serializable interface we do require that some thought be given by the programmer. Having serialization be the default state of an object would mean that lack of thought could cause bad effects in a program, something that the overall design of the Java platform has attempted to avoid.

source: http://java.sun.com/products/jdk/1.2/docs/guide/rmi/faq.html

-

17. Security

-

- **Question:** I'm developing an application to process a file. After the process, I have to crypt this file with a public key of a PGP user. I do not want to use exe file from within of Java application. Is it something was done with Java?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Yes, you can check one Java implementation here:
http://cryptix.org/products/openpgp/index.html

Cryptix OpenPGP is a Java implementation of the Open Specification for Pretty Good Privacy (OpenPGP), as specified in RFC 2440. It is compatible with popular programs like NAI's PGP and GnuPG.

Since it is an open project you can get access to source code.

- **Question:** I want to be able to uniquely identify a file by its contents. I read the file and use java.utils.zip.CRC32 to compute the checksum value.

  Does exist a possibility to have two files with different contents that return the same checksum?

  Answer: CRC32 generates a 32-bit checksum. There are only 2^32 possible values. So when you have more than 2^32 files you're guaranteed to have a collision.

  The possible solution to increase security could be to use Message Digest - technology based on MD5 or SHA.

  Message digests are secure one-way hash functions that take arbitrary-sized data and output a fixed-length hash value.

  The size of Message digests is always the same, independent of the size or content of the message from which it was created - 1024 bits.

  Of course, possibility exists that two files can have the same Message Digests as well.

  But the probability that two different files have the same checksums with CRC32 and the same message digests is negligibly low and probably (who can disprove?) equals zero.

  "Bullet-proof" approach should be:

  1. Check for CRC32 checksum for two files
  2. If they are the same check the message digests...

  More about Message Digests here:
  http://java.sun.com/j2se/1.4.1/docs/api/

- **Question:** Is it possible to use SSL with RMI and can I have both (plain and encrypted) connections on one JVM?

  Answer: Yes, it is possible, due to RMI enhancements available in the Java 2 SDK, Standard Edition, v1.2 (Java 2 SDK).

  The version of RMI included in the Java 2 SDK enables the RMI developer to use custom socket types. In earlier versions of RMI there could only be one type of

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

socket used per Java virtual machine* (JVM), but now RMI allows each object to use a different type of socket for its RMI connections. All an application has to do is install an RMI socket factory that creates sockets of the desired type (that speak a particular protocol).

Check tutorial on creating a custom RMI socket factory here:

http://java.sun.com/j2se/1.3/docs/guide/rmi/rmisocketfactory.doc.html

- **Question:** I am going to use "a code obfuscation" to prevent it from stealing. Are there some recommendations to make my code even more protected? Part 1

Answer: Yes. Exist a few code obfuscation products, one of them is JProof.

Personally, I think that this product is one of the best tools that you can find today.

To achieve maximum bytecode protection level you can use several techniques that allow to do better protection with JProof:

1. Make your methods and fields as much private as possible.

Usually a lot of stuff does not need to be accessed from the other classes. The good style is to mark such stuff as private (by adding "private" modifier to the field or method declaration). Additionally, this will little bit improve your application's performance.

2. If you can not make field or method private, try to make it "package".

Simply do not use any of public, private or protected modifiers in field or method declaration statement, and they automatically will be marked as "package" by Java compiler. Note that both public and protected members can be referenced from the other packages.

3. Minimize inter-package references.

All of the inter-package interaction should be performed via public interfaces and public classes.

4. If some classes in package do not referenced from other packages, remove the "public" modifier from their declarations.

In this case, Java compiler will mark these classes as "package", and JProof will be able to rename them and hide their actual names. Note that non-public class can implement public interfaces and publish some methods.

- **Question:** I am going to use "a code obfuscation" to prevent it from stealing. Are there some recommendations to make my code even more protected? Part 2

5. Change direct field access to set/get methods access.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Mark a field as private and add two methods for getting and setting field. This will improve protection within the class. You can leave usual (direct) field access for "package" (no-modifier) fields.

6. Implement "grouped" (joint) get/set methods where performance is not critical and where you need to make the best protection.

To do this, add new method with one additional int parameter that will indicate which exactly action need to be performed. Mark methods called from within it as private. This will reduce the number of public methods. Sometimes this technique is called as

"join methods". It is possible to join methods with similar signatures (not parameter names or sense) and add int "FnCode" parameter to new method's signature.

7. Use resource bundles to hold string constants.

This will improve the localizability of your software and will make more difficult human reverse-engineering.

8. Try to decompile your software yourself with all known Java decompilers.

This will help you to ensure that you are protected from most of Java decompilers. You can find a list of Java decompilers in the Decompilers section of "More Info" FAQ.

You can find this tool at: http://www.jproof.com/index.html

- **Question:** I used JProof free utility for code obfuscation to protect my code. It seems that application does not work properly...

  Answer: Yes, you should be aware that code obfuscation in certain areas will cause troubles.

  There are few cases when application's behavior may be affected:

  When Reflection API functions used to access private and package fields and methods of any class;

  When package classes accessed by name using

  Class.forName(String)

  call, and name of class to be loaded generated at run time from several parts;

  When changed Java items (classes, fields or methods) are referenced from native code.

Question: Can you help with third party pure Java programming language implementations of SSL?

Answer:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Phaos Technology's SSLava Toolkit:

http://phaos.com/home.html

Isasilk toolkit:

http://jce.iaik.tugraz.at/products/02_isasilk/index.php

Baltimore Technologies' J/SSL:

http://www.baltimore.ie/keytools/ssl/index.asp

Of course you can use SUN's implementation - a stand-alone Java programming language implementation of SSL for the Java 2 platform, the Java Secure Socket Extension (JSSE) 1.0:

http://java.sun.com/products/jsse/

Since Java 1.4, JSSE has been integrated into the Java 2 SDK, Standard Edition, and v1.4.

Ł

Ł

18. Servlets, Servers, Mail etc.

Ł

Ł **Question:** How can I measure performance of my servlets/JSP pages?

Answer: JMeter from jakarta.apache.org is simple, free and efficient.
There are several more from simplest
http://www.servletsuite.com/servlets/stress.htm

Stress tool for your web applications: lets you to test your servlets/JSP pages, supports session simulation, so you may test your real applications.

How does it work? Describe your host (set URI), define if you need sessions simulation (through cookies or URL rewriting), set parameters in case you are using post request and set hits intensity. Package simulates users activity based on the Puasson distribution. Intensity you set is an average amount hits per second (it may be non-integer, e.g. 0.5 means 30 hits per minute etc.). With the time step one second within the model time interval package calculates visitors count for this interval and generates a bundle of requests.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

What you will get?

Statistics for successful and failed requests and raw file with data for the each request: when it was (model time) and what was response time (real time or special mark for failed requests).

Package Stress distributed as a one jar file. Main class is StressTest. So - add jar file to your class path and run Stress.StressTest. Class requires one parameter - text file described your test configuration.
http://www.servletsuite.com/servlets/stress.htm

- **Question:** A serversocket call accept(). Does it listen the port forever until get a request? What's the meaning of non-blocking socket?

  Answer: When accept() is called the thread is blocked until either a connection is made or socket's timeout is expired.

  With non-blocking socket thread gets return straight away, and it is up to your application to decide what it is going to do - to stop or continue to check until a connection is established.

- **Question:** Are any classes available to deal with Mac Address (not IP) and which are these classes?

  Answer: Java deals at the higher level with protocol TCP/IP (layers 4/3). Mac address is used in layer 2 by Ethernet. Although you can work with them if you use ARP, RARP (Address Resolution Protocol and Reverse Address Resolution Protocol) and some native code...

- **Question:** How do I attach a file to the HTTP POST request? I am using URLConnection to send the HTTP request.

  Answer: you need to write code like this. Please find exact details yourself :-)

```
URL dest = "http://www.myurlhere.com ";
URLConnection urlCon = dest.openConnection();

// prepare input and output
urlCon.setDoInput(true);
urlCon.setDoOutput(true);

// Disable caching
urlCon.setUseCaches(false);

// Post output
DataOutputStream out = new DataOutputStream(urlCon.getOutputStream());
out.writeBytes( file.toString );
out.flush();
out.close();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
// return servlet response as a DataInputStream
DataInputStream in = new DataInputStream(urlCon.getInputStream());

// and so on
....
```

- **Question:** What are most used authentication methods for web applications?

  Answer: A few most common methods are used for authentication today:

  -basic, form based, digest and certificate methods.

  With basic authentication name and password are taken from login dialog.

  The weak side of this method is that the username and password are transmitted to the server using base64 encoding (that is, almost in plain text).

  The form-based authentication method is the same as a basic, but the server uses a custom login page, instead of displaying the standard login page. Username and password are sent in the same way as in method above.

  In contrast, the digest authentication method doesn't pass the password over the network at all. A series of numbers is generated (based on the password and other information about the request), and these numbers are then hashed using MD5 (Message Digest Algorithm). The hash value (digest) is sent over the network along with other information that the server can use to verify the password. This method is obviously more secure over the network.

  Client certificate authentication uses SSL (secure socket layer) and client certification. Developed by Netscape and become most popular authentication method that used on Internet.

- **Question:** How to do a POST Method from a java application instead of an html page?

  Answer: Use com.oreilly.servlet.HttpMessage from O'Reilly, Java Servlet Programming book:

  This class simplifies HTTP applet-server communication. It abstracts the communication into messages, which can be either

  GET or POST.

  It can be used like this:

```
URL url = new URL(getCodeBase(), "/servlet/ServletName");
HttpMessage msg = new HttpMessage(url);

// Parameters may optionally be set using java.util.Properties
Properties props = new Properties();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
props.put("name", "value");

// Headers, cookies, and authorization may be set as well
msg.setHeader("Accept", "image/png"); // optional
msg.setCookie("JSESSIONID", "9585155923883872"); // optional
msg.setAuthorization("guest", "try2gueSS"); // optional
InputStream in = msg.sendGetMessage(props);
```

Ł **Question:** I have a servlet with this code fragment:

```
response.setHeader("Cache-Control","no-cache");
response.setHeader("Pragma","no-cache");
response.setDateHeader ("Expires", 0);
```

But IE in any case shows and shows me the cached page...

What is wrong?

Answer: Microsoft writes here:

http://support.microsoft.com/default.aspx?scid=kb;EN-US;q234067

If the client communicates with the server over a secure connection (https://) and the server returns a Pragma: no-cache header with the response, Internet Explorer does not cache the response.

Pragma: no-cache prevents caching only when used over a secure connection. A Pragma: no-cache META tag is treated identically to Expires: -1 if used in a non-secure page. The page will be cached but marked as immediately expired.

When you use the <HTTP-EQUIV="PRAGMA" CONTENT="NO-CACHE"> metatag in the header section at the beginning of an HTML Web page, the Web page may still be cached in the Temporary Internet Files folder.

To properly prevent the Web page from appearing in the cache, place another header section at the end of the HTML document.

For example:

```
<HTML>
<HEAD>
<META HTTP-EQUIV="REFRESH" CONTENT="5">
<TITLE> Pragma No-cache </TITLE>
</HEAD>
<BODY>

This is an example of where to place the second header section<br>
so that the "Pragma, No-Cache" metatag will work as it is supposed to.<br>

</BODY>
<HEAD>
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
<META HTTP-EQUIV="PRAGMA" CONTENT="NO-CACHE">
</HEAD>
```

Cache-Control META HTTP-EQUIV tags are ignored and have no effect in Internet Explorer versions 4 or 5.

For Netscape the following JavaScript must be used in the BODY flag of all pages that should not be cached:

```
onLoad="if ('Navigator' == navigator.appName) document.forms[0].reset();"
```

- **Question:** How do I attach a file to the HTTP POST request? I am using URLConnection to send the HTTP request.

  Answer: you need to write code like this. Please find exact details yourself :-)

```
URL dest = "http://www.myurlhere.com ";
URLConnection urlCon = dest.openConnection();

// prepare input and output
urlCon.setDoInput(true);
urlCon.setDoOutput(true);

// Disable caching
urlCon.setUseCaches(false);

// Post output
DataOutputStream out = new DataOutputStream(urlCon.getOutputStream());
out.writeBytes( file.toString );
out.flush();
out.close();

// return servlet response as a DataInputStream
DataInputStream in = new DataInputStream(urlCon.getInputStream());

// and so on
....
```

- **Question:** Is it possible to run a servlet from a browser without visibly changing the browser address or opening a new browser window?

  Answer: Yes it is possible if the servlet return HTTP status code 204 (No Content). This tells the browser to continue displaying the previous document.

- **Question:** Can I run a Java application through a web page? Can be a servlet used for this purpose or it can be done in some another way?

  Answer: Java Web Start -- a technology for simplifying deployment of Java applications -- gives you the power to launch full featured applications with a single click from your Web browser. You can now download and launch applications, such

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

as a complete spreadsheet program or an Internet chat client, without going through complicated installation procedures.

Java Web Start includes the security features of the Java 2 platform, so the integrity of your data and files is never compromised.

In addition, Java Web Start technology enables you to use the latest Java 2 technology -- with any browser.

With Java Web Start, you launch applications simply by clicking on a Web page link. If the application is not present on your computer, Java Web Start automatically downloads all necessary files. It then caches the files on your computer so the application is always ready to be relaunched anytime you want -- either from an icon on your desktop or from the browser link.

And no matter which method you use to launch the application, the most current version of the application is always presented to you. Java Web Start is localized into French, German, Italian, Japanese, Korean, Simplified Chinese, Spanish, Swedish, and Traditional Chinese.

Java Web Start is easy to use: you can efficiently access, download, and launch applications as easily as accessing a Web page.

Java Web Start 1.2 is now available for download as part of J2SETM. Beginning with version 1.2, Java Web Start will be bundled with J2SE and will no longer be a separate download.

Because Java Web Start runs independently of any browser, you can shut down your browser or surf to another Web page and continue using your application. Applications deployed with Java Web Start do not require a browser interface to operate. To save time later, you can relaunch your desired application anytime from the desktop icon, without powering up your browser again.

- **Question:** I have written a small Java web server.

  I want to delete (replace) a class that was loaded with Class.forName("name"). I have not found a way to do that. Each time I get an old class...

  Answer: You need to replace the Class.forName() with ClassLoader.loadClass() and instantiate a new ClassLoader every time you need a new version of an existing class.

- **Question:** I see this error message:

  java.net.SocketException:

  Connection reset by peer: socket write error even when I use Tomcat as a localhost server.

  Answer: This behaviour results from HTTP protocol. Each new page request results in a new TCP connection. When a user press STOP or RELOAD button or goes to

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

another page the browser resets the old connection resulting this exception on server side.

Do not worry about, just properly handle it!

- **Question:** I'm trying to read a file from the JSP file on the server. It works fine when I try to read the same file on my PC with the following statement:
FileReader f = new FileReader("d:\temp\a.txt");
BufferedReader fi = new BufferedReader(f);

  However, when I run the same JSP file with the same statement on the server , I get the exception saying that the file cannot be found...
I am sure that the file is in the same directory as the JSP file that tries to open it.

  FileReader f = new FileReader("a.txt");
BufferedReader fi = new BufferedReader(f);

  Here is the error when the above is executed by JSP on the server:

  500 Internal Exception
error = java.io.FileNotFoundException: a.txt

  Answer: The JSP is not in the same directory as the text file,
The JSP is stored in a default location within the server itself because it is compiled to a servlet.

  You have two options:

  1. Find this location and put your text file there,

  or

  2. Provide the full path to file reader the full path or give it a relative path (the best choice, since the code will be portable.

- **Question:** The first thing is that i don't know how to do to create Servlets ... no more precisely, I don't know how to compile it!!

  It says that it doesn't find the packages (javax I think) ...
Plz, could someone tell me exactly what I should put on my environment variables (Class_path, path, java_home etc...)

  Answer: Well, I also started to learn about servlets. I recently downloaded JDK1.3.0 and Apache Tomcat 3.1.0.
I set the following paths.

  CLASSPATH = %TOMCAT_HOME%\webapps\examples\WEB-INF\classes\work;
    (my working directory)
TOMCAT_HOME = C:\JDK\tomcat;
    (or wherever you installed tomcat)
Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

PATH=%TOMCAT_HOME%\bin ;
appended tomcat bin path to the path statement)

I also copied the servlet.jar found in the tomcat bin directory to both:

C:\jdk\jre\lib\ext\
C:\Program Files\JavaSoft\JRE\1.3\lib\ext\

I hope this helps....

Gregory Pedder

- **Question:** I do not see constructors in servlets? Am I missing something? How can servlet run without constructor?

  Answer: There is no need to use constructor in servlets for you because servlet are similar to applet in a way it is running. The creation of servlet instance is a Server responsibility. You just need to use init() method.
  This method called first when servlet is loaded into servlet container...
  --
  AP (J.A.)

- **Question:** Where can I find online books about servlets?

  Answer: I have on the site free books link collection. Doing search for "servlet" I got 5 books. Please check here:
  http://www.javafaq.nu/indexu/search.php4?keyword=servlet
  --
  John

- **Question:** When can I use System.exit() in servlets?

  Answer: Never! Depends on server you run on... Security exceptions on good server or full shut down for simpler one.
  --
  John

- **Question:** Why Servlets are better than CGI scripts?

  Answer: Generally, every new client's request to CGI script starting a new process. With servlet web server starts just a new thread that is much cheaper in terms of time for starting up, memory and CPU consumption.
  JVM uses just one process.
  This answer is right for "standard" CGI. Now appear new modules for Perl that uses the same approach as servlet does. Anyway, writing large programs on Java is much easier than on Perl, at least for me.
  Java has also very rich API.
  --
  John

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I was writing and testing some servlet program. Now I decided to remove it from our web server.
But it is still there causing confusion to another developers I am working with.
How can I unload my servlet explicitly?

  Answer: It depends on server you use. Often you can unregister your servlet somehow.
Easier way to make your servlet empty (remove all code), compile it and copy class file to server.
Many servers reload automatically a new class if they see it was changed. You will still have your servlet on the server but it will do nothing and nobody will complain...
--
Peter

- **Question:** Any simple server I can download to run my java servlets...

  Answer:
Try Resin, www.caucho.com
Try tomcat from http://jakarta.apache.org - pure Java,
servlets 2.2, good stuff! Tomcat is good. It's not fast, but it's very easy to setup and good solution for development. Since most heay-duty servers do implement SUN specification, one can migrate application in no time.
--
DG

  Jigsaw is also good but doesn't support the Servlet 2.2 spec

  <URL:http://jigsaw.w3.org/>

  Both of these are open source with BSD-like license.

  There's also Jetty http://jetty.mortbay.com/ Artistic license, Servlet 2.2 spec compliant), GNU Paperclips
http://www.tapsellferrier.co.uk/gnupaperclips/ GPL, claims to be moving towards Servlet 2.3 spec compliance) and also
vqServer http://www.vqsoft.com/vq/server/ which is free but not open source.

  Finally many of the commercial servers can be downloaded for free trial periods.
--
Simon Brooke http://www.jasmine.org.uk/~simon

- **Question:** Can someone point me to some tutorial/example/text book that covers the subject of receiving on a server side a file upload from a browser sent as with content-type set to multipart/form-data.
Thanks.

  Answer: There are free Java classes that will do this for you. Here's one I wrote:

  http://users.boone.net/wbrameld/multipartformdata/
If you still want to write your own, you can use mine as an example.

  or here:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

http://www.servlets.com/cos/index.html
There you can download package which contains also examples, how to do it.
--
Walter Brameld

- **Question:** How can I get access to Cookie set at the Client?

  Answer: The following code should access a cookie on a client. It reads in all the cookies on the machine. And checks there name for whichever one you are looking for.

```
Cookie[] cookies = request.getCookies();
for(int i=0; i < cookies.length; i++) {
   Cookie thisCookie = cookies[i];
   if (thisCookie.getName().equals("Cookiename")) {
        // Do whatever you want with the cookie....
   } else {
       // cookie doesn't exist...
   }
}
```

  The Cookie class is in package javax.servlet.http.Cookie

- **Question:** I'd like to provide to the showDocument() method of an applet the URL of a CGI program...

  I'd like to provide to the showDocument() method of an applet the URL of a CGI program with including a certain number of (URL-encoded) parameters to this URL (which is the same as doing a GET HTTP request).
  What is the maximum size I can give to this URL ?

  Answer: If I remember exactly it something around 240 for many servers. Maybe less, but not more!!!
  1000000%
  I read it last year in "Java Servlet Programming" from O'Reily.

- **Question:** I am experimenting with a Java server application. This program has worked well.

  It did start on the Red Hat Linux 6.0 server, but it does not open the socket, in other words, it cannot communicate with the client applet on the Linux. On this Linux server I have installed every component and all of them were running at the experiment time. Why does this server application communicate with the client applet only on the Linux? Does anyone give me a suggestion?

  Answer: Take a look at your port number. If it is under 1024, it is a protected port number and non-privileged users cannot touch it on Linux or any other Unix-system.

- **Question:** Where Can I find a server to try my servlets?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

I am creating a client/server application. I don't run my own server and my ISP won't allow me to install and run applications from their server.
Does anyone know of anywhere (preferably FREE) that will allow me to use server side Java? Any help is GREATLY appreciated.

Answer: http://www.mycgiserver.com/

- **Question:** Hi, I am using servlets. I need to store an object NOT a string in a cookie. Is that possible? The helpfile says BASE64 encoding is suggested for use with binary values. How can I do that???

Answer: You could serialize the object into a ByteArrayOutputStream and then Base64 encode the resulting byte []. Keep in mind the size limitations of a cookie and the overhead of transporting it back and forth between the browser and the server.
Limitations are:

* at most 300 cookies

* at most 4096 bytes per cookie (as measured by the characters that comprise the cookie non-terminal in the syntax description of the Set-Cookie2 header, and as received in the Set-Cookie2 header)

* at most 20 cookies per unique host or domain name

For more details please refer to RFC 2965.

- **Question:** Hi, I want to send a POST request, but I can't find such functionality in the servlet API, how can I do this? Must I implement this with a socket connection to port 80?

Answer: A servlet can do anything a standalone Java application can do. It doesn't need anything beyond what the java.net package already provides. You can use an httpURLConnection to POST to a server program like a servlet or CGI script:

```
// Create a string with the parms you want to post  and convert it to a byte array.
You may need to
// pass the values through java.net.URLEncoder.encodeURL()
// if they have embedded blanks or special characters

String parms = "a=10" + "&b=20" + "&c=30";
byte[] bytes = parms.getBytes();

// Create a URL pointing to the servlet or CGI script and open an HttpURLConnection
on that URL

URL url = new URL(TARGET_URL);
HttpURLConnection con = (HttpURLConnection) url.openConnection();

// Indicate that you will be doing input and output, that the method is POST, and
that the content
// length is the length of the byte array
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
con.setDoOutput(true);
con.setDoInput(true);
con.setRequestMethod("POST");
con.setRequestProperty("Content-length", String.valueOf(bytes.length));

// Write the parameters to the URL output stream

OutputStream out = con.getOutputStream();
out.write(bytes);
out.flush();
// Read the response
BufferedReader in = new BufferedReader(
            new InputStreamReader(con.getInputStream()));
  while (true) {
      String line = in.readLine();
      if (line == null) break;
      System.out.println(line);
  }
in.close();
out.close();
con.disconnect();
```

--
Phil Hanna
Author of Instant Java Servlets
http://www.philhanna.com

- **Question:** I am writing an application, using Java Servlets, which requires me to set and read cookies. It works okay, but fails if I set the domain of the cookie to something other than the current server.

  I am writing an application, using Java Servlets, which requires me to set and read cookies (the servlet API, and the javax.servlet.http.Cookie class).

  I created a new cookie, and added it to the http response, using its addCookie() method. It works okay, but fails if I use the setDomain method, on the newly created cookie, to set the domain of the cookie to something other than the current server.

  Answer: I suspect that is not a legal operation for any browser to accept a cookie that has a domain inconsistent with the source of the cookie.
  by William Brogden

- **Question:** I am working on weblogic server 5.1 with MsSQLSERVER7 I am able to load the driver but it says unable to get socket connection

  I am working on weblogic server 5.1 with MsSQLSERVER7 i am able to load the driver but it says unable to get socket connection. It says connect to the MSSQLSERVER's host and port no.
  How do I get these name and value.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: The MS SQL Server's host is usually the name or IP address of the server that run SQL Server, if you know the IP (ping <nameoftheserver>), put the IP in it, it will be faster, for the Port number, sincerely I don't remember the standard port, but look into the SQL Server documentation and you will find it.
by Davide

- **Question:** Whenever I compile my servlets it always says "can't find package javax.*" even though I downloaded the JSDK. Where the JSDK files so it'll find that package?

  Answer: There are no classes in the javax.* package. There are classes in javax.servlet.* and javax.servlet.http.*, but neither are really related to javax.* -- importing javax.* won't affect them. You should import the packages that you really want to use!

- **Question:** I have a list of html links on a web page, and I want to be able to call a servlet based on what the user clicked on...

  I used:

  <a href=/servlet/SomeServlet">

  for each link, and I want the servlet to display information based on what the user clicked on. If the user click on an applet link, I want the Servlet to print, "You just clicked on an applet", etc.

  My question is, how do I send information to a servlet, based on what html link the user clicked on? I know i can use getParamaterValue() for getting information off of forms, but I'm not sure how to do this with html tags.

  Answer: Change the link to:

  <A href="/servlet/SomeServlet?click=apple">link</A>

  In the servlet, use request.getParameter("click") to retrieve the value.

  Give each link a unique "click" value and that will tell you what was clicked.
  B Russell

- **Question:** I'm looking for a Java HTTP server framework that can be used and modified for a project I'm going to be working on.

  My boss asked me what the equivalent in Java was that IIS was in the Windows world or Apache was in the UNIX. I've looked at Jigsaw, but am wondering if anyone out there knows of other resources...open source would be great. Thanks in advance for any input.

  Answer: There are other pure Java web servers, like the Apache PicoServer, Jetty (http://www.jetty.org ???). Perhaps you could take one of those to write a http block for the Apache Server Framework avalon (http://java.apache.org/framework)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Another tip is to search sourceforge for web servers.
http://www.sourceforge.net
--
Glen Shelly

- **Question:** I am currently running Microsoft's IIS server. I have the Java Virtual machine installed on this system. What else will I need to run my servlet? Do need the Apache web server instead? Will IIS support Java servlet?

Answer: You will need a Servlet Engine to run your servlets.
You can use either Allaire's JRun (http://www.jrun.com) or Tomcat (http://jakarta.apache.org).
Both of them work with IIS.
--
Madhusudhanan Challur

- **Question:** How can I avoid browser caching?

In an applet, I call the function showDocument to send a GET message to one of my servlets, which sends me back a PDF file dynamically generated by this servlet.
My problem is : the browser (in this case IE) always sends me back a PDF file with the same content.

Answer 1: There are a few possibilities to avoid this problem.
A simple workaround is to add a random number URL variable with the request, such as "ranx" with a value of the milliseconds since midnight. This will make the request unique and, hence, avoid browser caching.
by Duane Morse

Answer 2: There are two other response setHeader attributes you can set:

```
response.setHeader("pragma", "no-cache");
response.setHeader("Cache-Control", "no-cache");
response.setDateHeader("Expires", 0 )
```

Just in case.
Doug Schwartz

Answer 3: // for Answer 1 and 2 please go to yesterday's tip.
When you generate the PDF file, make sure you set the header to tell the browser that the file will expire at a certain time. The browser should not cache the response past the given time. The Java code looks something like
this:

```
import java.text.SimpleDateFormat;
import java.util.Date;

SimpleDateFormat dateFormat
= new SimpleDateFormat("EEE, dd MMM yyyy - HH:mm:ss z");

response.setHeader("Expires", dateFormat.format(new Date()));
```

The format of the date is very particular.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Scott Gartner

- **Question:** Which free software can I use to debug Servlets?

Answer: What you need to do is use Jakarta-Tomcat to run your servlets. Then, bring in all the tomcat jars into your "required libraries".
Then set up your "project run" to actually run the Tomcat class. If you run it in debug mode, you should be able to put breakpoints in your servlets.

We did this at a project I was on with weblogic and EJBs, and JBuilder 4 does it with tomcat, so I'm assuming you can do this as well.\
--
Trever M. Shick
http://xjr.sourceforge.net
http://velocidoc.sourceforge.net
http://www.objectwave.com
http://www.geocities.com/trevershick

- **Question:** Is tomcat an EJB container?

Answer: No. It's only a servlet/JSP container. However, you can certainly use it with an EJB container, and you can get another free EJB container such as JBoss.
--
Chris Smith

-

-

19. Sound, Speech and Multimedia

-

-

- **Good to know:** Do you know that Version 1.3 of the Java TM 2 Platform includes a powerful new API for capturing, processing, and playing back audio and MIDI (Musical Instrument Digital Interface) data.
You can read here more: http://java.sun.com/j2se/1.3/docs/guide/sound/index.html

- **Question:** I'm doing a small console java app, and want to know what the function is to make the pc speaker beep....

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: Toolkit.getDefaultToolkit().beep();

Ł **Question:** I just starting with Java and going to develop a class that will give reverberation effect to my game. Where should I start - to create physics model or search for some already written code?

Answer: You should always start from Java API! It is not big secret that Java grows fast and while you, for example spending time with sockets programming, another guys spending theirs time on API improvement.

I advise you - always firstly look at Java API, secondly look at my tips :-)
http://javaFAQ.nu

Then look at outstanding articles of Dr. Kabutz (see announcement below)
And just after that go http://java.sun.com and finally http://google.com

Fortunately the reverberation class is already developed and can be found in Java since 1.3. It offers wide range of effects from garage to closet.

"Depending on the size of the room, and how absorbent or reflective the materials in the room's surfaces are, the sound might bounce around for a long time before dying away.

The reverberation parameters provided by ReverbType consist of the delay time and intensity of early reflections, the delay time and intensity of late reflections, and an overall decay time. Early reflections are the initial individual low-order reflections of the direct signal off the surfaces in the room. The late Reflections are the dense, high-order reflections that characterize the room's reverberation. The delay times for the start of these two reflection types give the listener a sense of the overall size and complexity of the room's shape and contents. The larger the room, the longer the reflection delay times. The early and late reflections' intensities define the gain (in decibels) of the reflected signals as compared to the direct signal. These intensities give the listener an impression of the absorptive nature of the surfaces and objects in the room. The decay time defines how long the reverberation takes to exponentially decay until it is no longer perceptible ("effective zero"). The larger and less absorbent the surfaces, the longer the decay time.

The set of parameters defined here may not include all aspects of reverberation as specified by some systems. For example, the Midi Manufacturer's Association (MMA) has an Interactive Audio Special Interest Group (IASIG), which has a 3-D Working Group that has defined a Level 2 Spec (I3DL2). I3DL2 supports filtering of reverberation and control of reverb density. These properties are not included in the JavaSound 1.0 definition of a reverb control. In such a case, the implementing system should either extend the defined reverb control to include additional parameters, or else interpret the system's additional capabilities in a way that fits the model described here.

If implementing JavaSound on a I3DL2-compliant device:

-Filtering is disabled (high-frequency attenuations are set to 0.0 dB)
Density parameters are set to midway between minimum and maximum"

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The table that shows what parameter values an implementation might use can be found in Java API.

Question: How Does the Java Sound API Relate to Other Interfaces?

Answer: The Java Sound API provides the lowest level of sound support on the Java platform. It provides application programs with a great amount of control over sound operations, and it is extensible. For example, the Java Sound API supplies mechanisms for installing, accessing, and manipulating system resources such as audio mixers, MIDI synthesizers, other audio or MIDI devices, file readers and writers, and sound format converters. The Java Sound API does not include sophisticated sound editors or graphical tools, but it provides capabilities upon which such programs can be built. It emphasizes low-level control beyond that commonly expected by the end user.

There are other Java platform APIs that have sound-related elements. The Java Media Framework (JMF) is a higher-level API that is currently available as a Standard Extension to the Java platform. JMF specifies a unified architecture, messaging protocol, and programming interface for capturing and playing back time-based media. JMF provides a simpler solution for basic media-player application programs, and it enables synchronization between different media types, such as audio and video. On the other hand, programs that focus on sound can benefit from the Java Sound API, especially if they require more advanced features, such as the ability to carefully control buffered audio playback or directly manipulate a MIDI synthesizer. Other Java APIs with sound aspects include Java 3D and APIs for telephony and speech. An implementation of any of these APIs might use an implementation of the Java Sound API internally, but is not required to do so.

/from the Java guide

Question: How can I be sure that unknown to me applets or Java applications are not recording sound from a microphone in my PC and do not send records out?

Answer: The Java Sound API includes an AudioPermission class that indicates what kinds of access an applet (or an application running with a security manager) can have to the sampled-audio system. Permission to record sound is controlled separately. This permission should be granted with care, to help prevent security risks such as unauthorized eavesdropping. By default, applets and applications are granted permissions as follows:

- An applet running with the applet security manager can play, but not record, audio.
- An application running with no security manager can both play and record audio.
- An application running with the default security manager can play, but not record, audio.

In general, applets are run under the scrutiny of a security manager and aren't permitted to record sound. Applications, on the other hand, don't automatically install a security manager, and are able to record sound. (However, if the default security manager is invoked explicitly for an application, the application isn't permitted to record sound.)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Both applets and applications can record sound even when running with a security manager if they have been granted explicit permission to do so.

/from the Java guide

Question: Does Java support memory buffer optimizations to speed up video?

Answer: The BufferStrategy class represents the mechanism with which to organize complex memory on a particular Canvas or Window. Hardware and software limitations determine whether and how a particular buffer strategy can be implemented. These limitations are detectible through the capabilities of the GraphicsConfiguration used when creating the Canvas or Window.

It is worth noting that the terms buffer and surface are meant to be synonymous: an area of contiguous memory, either in video device memory or in system memory.

There are several types of complex buffer strategies; sequential ring buffering, blit buffering, and stereo buffering are common types. Sequential ring buffering (i.e., double or triple buffering) is the most common; an application draws to a single back buffer and then moves the contents to the front (display) in a single step, either by copying the data or moving the video pointer. Moving the video pointer exchanges the buffers so that the first buffer drawn becomes the front buffer, or what is currently displayed on the device; this is called page flipping.

Alternatively, the contents of the back buffer can be copied, or blitted forward in a chain instead of moving the video pointer.

Stereo buffering is for hardware that supports rendering separate images for a left and right eye. It is similar to sequential ring buffering, but there are two buffer chains, one for each eye. Both buffer chains flip simultaneously.

From Java API for BufferStrategy class

- **Question:** I would like to burn a CD from within my code, that way I won't have to waste so much time making illegal copies of music. Is there a convenient way to do this in Java?

  Answer: Unfortunately Java doesn't provide any API for this :-)

- **Question:** I tried to look at the classes for Java Speech technology... They are not included into JDK. Even more, it seems SUN has no implementation for it. How we can use it if there are no classes? Part 1

  Answer: Yes, that's right. Sun does not ship an implementation of JSAPI. Instead, it works with third party speech companies.

  JSAPI (Java Speech API) allows Java applications to incorporate speech technology into their user interfaces. It defines a cross-platform API to support command and control recognizers, dictation systems and speech synthesizers.

  JSAPI includes the Javadoc-style API documentation for the approximately 70 classes and interfaces in the API. The specification also includes a detailed Programmer's

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Guide which explains both introductory and advanced speech application programming with JSAPI.

Two companion specifications are available: JSML and JSGF:

The Java Speech API Markup Language (JSML) and the Java Speech API Grammar Format (JSGF) are companion specifications to the Java Speech API. JSML (currently in beta) defines a standard text format for marking up text for input to a speech synthesizer.

JSGF version 1.0 defines a standard text format for providing a grammar to a speech recognizer. Both specifications are available at http://java.sun.com/products/java-media/speech/

- **Question:** I tried to look at the classes for Java Speech technology... They are not included into JDK. Even more, it seems SUN has no implementation for it. How we can use it if there are no classes? Part 2.

The Java Speech API is a freely available specification and therefore anyone is welcome to develop an implementation. The following implementations are known to exist:

FreeTTS on SourceForge.net

Description: Open source speech synthesizer written entirely in the Java programming language.
Requirements: JDK 1.4. Read about more requirements on the FreeTTS web site.

IBM's "Speech for Java"

Description: Implementation based on IBM's ViaVoice product, which supports continuous dictation, command and control and speech synthesis. It supports all the European language versions of ViaVoice -- US & UK English, French, German, Italian and Spanish -- plus Japanese.
Requirements: JDK 1.1.7 or later or JDK 1.2 on Windows 95 with 32MB, or Windows NT with 48MB. Both platforms also require an installation ViaVoice 98.

IBM's "Speech for Java" on Linux

Description: Beta version of "Speech for Java" on Linux. Currently only supports speech recognition.
Requirements: RedHat Linux 6.0 with 32MB, and Blackdown JDK 1.1.7 with native thread support.

The Cloud Garden

Description: Implementation for use with any recognition/TTS speech engine compliant with Microsoft's SAPI5 (with SAPI4 support for TTS engines only). An additional package allows redirection of audio data to/from Files, Lines and remote clients (using the javax.sound.sampled package). Some examples demonstrate its use in applets in Netscape™ and IE browsers.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Requirements: JDK™ 1.1 or better, Windows 98, Me, 2000 or NT, and any SAPI 5.1, 5.0 or 4.0 compliant speech engine (some of which can be downloaded from Microsoft's web site).

- **Question:** I tried to look at the classes for Java Speech technology... They are not included into JDK. Even more, it seems SUN has no implementation for it. How we can use it if there are no classes? Part 3.

Lernout & Hauspie's TTS for Java Speech API

Description: Implementations based upon ASR1600 and TTS3000 engines, which support command and control and speech synthesis. Supports 10 different voices and associated whispering voices for the English language. Provides control for pitch, pitch range, speaking rate, and volume.
Requirements: Sun Solaris OS version 2.4 or later, JDK 1.1.5. Sun Swing package (free download) for graphical Type-n-Talk demo.
More information: Contact Edmund Kwan, Director of Sales, Western Region Speech and Language Technologies and Solutions (ekwan@lhs.com)

Conversa Web 3.0

Description: Conversa Web is a voice-enabled Web browser that provides a range of facilities for voice-navigation of the web by speech recognition and text-to-speech. The developers of Conversa Web chose to write a JSAPI implementation for the speech support.
Requirements: Windows 95/98 or NT 4.0 running on Intel Pentium 166 MHz processor or faster (or equivalent). Minimum of 32 MB RAM (64 MB recommended). Multimedia system: sound card and speakers. Microsoft Internet Explorer 4.0 or higher.

Festival

Description: Festival is a general multi-lingual speech synthesis system developed by the Centre for Speech Technology Research at the University of Edinburgh. It offers a full text to speech system with various APIs, as well an environment for development and research of speech synthesis techniques. It is written in C++ with a Scheme-based command interpreter for general control and provides a binding to the Java Speech API. Supports the English (British and American), Spanish and Welsh languages.
Requirements: Festival runs on Suns (SunOS and Solaris), FreeBSD, Linux, SGIs, HPs and DEC Alphas and is portable to other Unix machines. Preliminary support is available for Windows 95 and NT. For details and requirements see the Festival download page.

- **Question:** I tried to look at the classes for Java Speech technology... They are not included into JDK. Even more, it seems SUN has no implementation for it. How we can use it if there are no classes? Part 4.

Elan Speech Cube

Description: Elan Speech Cube is a Multilingual, multichannel, cross-operating system text-to-speech software component for client-server architecture. Speech Cube is available with 2 TTS technologies (Elan Tempo : diphone concatenation and

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Elan Sayso : unit selection), covering 11 languages. Speech Cube native Java client supports JSAPI/JSML.
Requirements: JDK 1.3 or later on Windows NT/2000/XP, Linux or Solaris 2.7/2.8, Speech Cube V4.2 and higher.
About Elan Speech: Elan Speech is an established worldwide provider of text-to-speech technology (TTS). Elan TTS transforms any IT generated text into speech and reads it out loud.

- **Question:** What audio and video formats does the Java Media Framework support?

Answer: JMF supports a lot of different media formats. I think it better to give you a link instead of listing them here. The list grows and can be obsolete for those who read my tip later. So, for Java Media Framework 2.1.1 (latest version today) it is here:

http://java.sun.com/products/java-media/jmf/2.1.1/formats.html

- **Question:** I use System.out.println("\007") to make a beep. It works fine with java.exe version 2.2.2_007, but doesn't work when I use javaw.

It's said that java and javaw are exactly the same but apparently they have some differences.
How I can use System.out.println("\007") to make a beep with javaw?

Answer: System.out.println sends a character to the standard output stream, which is redirected to nowhere when you use javaw. Same would happen if you ran your application under UNIX and redirected standard out to /dev/null. I'd guess that your code wouldn't work at all on a Macintosh.

To solve this problem, do what you need more directly, for example using java.awt.Toolkit.beep instead of a non-portable kludge with sending special ASCII codes to an output stream.

Chris Smith

- **Question:** Just wondering if Java has classes and methods supporting joysticks? Is there any sample code working with joysticks some where?

Answer: Sure it does. It's part of the Java 3D API. There is an InputDevice interface. It is described as follows:

"InputDevice is the interface through which Java 3D and Java 3D application programs communicate with a device driver. All input devices that Java 3D uses must implement the InputDevice interface and be registered with Java 3D via a call to PhysicalEnvironment.addInputDevice(InputDevice). An input device transfers information to the Java 3D implementation and Java 3D applications by writing transform information to sensors that the device driver has created and manages. The driver can update its sensor information each time the pollAndProcessInput method is called."

The trick is finding a driver :-)

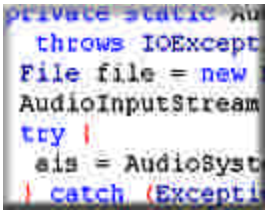
Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

http://sourceforge.net/projects/j3djoystick/
http://www.j3d.org/utilities/sensors.html

Jim S.

20. String, text, numbers, I/O



Question: What is difference between:

String My = "Test";
and
String My = new String ("My"); ?

Answer: Because the compiler automatically creates a new String object for every literal string it encounters, you can use a literal string to initialize a String.
String s = "Test";

The above construct is equivalent to, but more efficient than, this one, which ends up creating two Strings instead of one:
String s = new String("Test");

The compiler creates the first string when it encounters the literal string "Test", and the second one when it encounters new String.

source: http://java.sun.com/docs/books/tutorial/java/data/stringsAndJavac.html

Question: I need to program "xyz" math function. Where can I find code examples?

Answer: Big code examples collection can be found at:http://source42.netfirms.com/
It includes:
-Advance Math Functions
-Combinatorics
-Complex Numbers
-Crypt
-Graphics
-Graphs
-Math Functions and Simple Algorithms

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- Masc...
- Searching
- Sorting
- Statistics
- Strings
- Trees
- Vector And Matrix

Ł **Question:** How can I convert this string 23AB to its hexadecimal representation 32334142.

Is there any method that can do that easily? Or how could I write this function?

Answer: It is easy indeed:

Integer.parseInt(String s, int radix)

(for hexadecimal radix=16)

and

Integer.toHexString(int i)

Question: What is difference between JNLP (Java Network Launching Protocol) and JINI?

Answer: On the face of it, Jini and JNLP appear to occupy similar spaces in that they both allow code to be migrated to a client machine and to execute there. The following table summarises some of the differences and similarities:

Jini downloads a service

JNLP downloads an application

Jini: a client must be running to request a service

JNLP: a browser must be running which calls a helper to start the application

Jini: each service looks after itself, independently of any clients

JNLP: each JNLP file specifies all required parts of an application; if any part changes, the JNLP file must be updated

Jini: the client may need to know the location of a lookup service, but not of any service

JNLP: the user of a JNLP application must know the URL of the JNLP file

Jini: no generic client

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

JNLP: generic JNLP helper

Jini and JNLP Comparison is done in the book by Jan Newmarch,

"A Programmer's Guide to Jini Technology"
http://pandonia.canberra.edu.au/java/jini/tutorial/JNLP.xml

Question: When can I use Jini and when JNLP? Can I combine them together?

Part 1

Answer: Good answer was found in the book of Jan Newmarch,

"A Programmer's Guide to Jini Technology".

http://pandonia.canberra.edu.au/java/jini/tutorial/JNLP.xml#Jini%20and%20JNLP%20Comparis

By the way this book I am going to publish on my site soon!

"If a user does not have the client-side of a Jini application installed, then they cannot make use of Jini services. Here is where JNLP can help, by allowing a user to download an application that can run as the client. The user only needs to know a URL for this, and finding URLs is a common experience for most users nowadays. Corporate Web sites, Web search engines, portals and so on are all mechanisms used to find interesting Web sites and download information.

The converse question is: if JNLP is used to find an application, what is the value of Jini in this? The answer to this lies in the distributed management of Jini services. Suppose a JNLP application relies on a number of components put together, say as a collection of packages. The JNLP file has to specify the location of every one of these packages. If one of the packages changes, then the JNLP file has to be updated. It gets more complex if one of the packages changes to become dependant upon another new package: that new package has to be added to the JNLP file. In other words, management of a JNLP application has to be centralised, to the manager of the JNLP file.

Jini, on the other hand, lets every package/service be managed by its own manager. If a Jini service changes implementation, then it can do so without any external consultation, and just re-registers the new implementation with Jini lookup services. If a service changes to use another service, it does not need to inform anyone else about this change. A Jini client does not need to know how services are implemented or even where they are located.

Question: When can I use Jini and when JNLP? Can I combine them together?

Part 1

Jini is not quite management-free: a client may need to know where lookup services reside. On a local network a client can use multicast to locate lookup services, but outside of this local network clients will need to use unicast to find lookup services at known locations. This is still an improvement: Jini lookup services are relatively

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

stable, persistent and stationary services, whereas the services themselves may be transient or unstable.

The combination of Jini and JNLP works like this:

A user finds the URL for an application

The user downloads the JNLP file which is executed by the JNLP helper

The helper downloads the application's jar files, and runs a Java runtime engine with these jar files

Properties are set in the JNLP file for Jini lookup services, so the application can locate its required Jini lookup services at runtime

When the application needs a service, it finds it from the Jini lookup services and uses it

If the service needs to make use of other services, then it can perform its own search for them, without the knowledge of the application."

Question: Why my PrintStream never throws an IOException?

Answer: Unlike other output streams, a PrintStream and a PrintWriter never throws an IOException; instead, exceptional situations merely set an internal flag that can be tested via the checkError method.

Question: Does PrintWriter always automatically flash their output? Why in the constructor second value is "boolean autoFlush"

Answer: PrintStream and PrintWriter objects do not always flush their output. To arrange for automatic flushing, use the two-argument constructors of these classes and specify true for the second argument.

Optionally, a PrintStream can be created so as to flush automatically; this means that the flush method is automatically invoked after a byte array is written, one of the println methods is invoked, or a newline character or byte ('\n') is written.

All characters printed by a PrintStream are converted into bytes using the platform's default character encoding. The PrintWriter class should be used in situations that require writing characters rather than bytes.

Good to know: Do you know that The read(byte[]) and read(byte[], int, int) methods of InputStream and its subclasses are not guaranteed to read all available bytes. A loop may be required, for example, in order to read a large file into an array:

```
for (int off = 0; off < size;) {

    int r = in.read(buf, off, buf.length - off);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
        if (r == -1) break;

        off += r;

    }
```

Alternatively, a BufferedInputStream may be used. Similar remarks apply to the read(char[]) and read(char[], int, int) methods of Reader and its subclasses.

Question: I want to have a listener for a String's change event. Is it possible?

Answer: String has no such listener because you can not change a String.

Strictly saying a String can not be changed. Each time when you change a value of an "old" String Java creates a new String and

then refers to "old" one.

However you can write a wrapper class which has two methods - setString() and getString(). The body of those two methods could have additional code that notifies about String's "change" event. Something like this:

```
public void setString (String aString) {
  this.aString = aString;
  stringValueChanged();
}
```

The method stringValueChanged() is responsible for notification

Question: I just started with Java and found very strange that in simple operation like:

```
float a = 31.335f;
float b = 12.2362f;
float result = a + b
gives
43.571200000000005 not 43.5712
Why?
```

Even more:
if I do simple operation

```
d = a + b;
d = d - b;
```

it will not give exactly the value of "a":
31.335 not equal to 31.334997?

Answer:
The reason is that computers usually use counting based on base 2, not 10 and if we use float variables then they often can not be exactly converted to binary number. For doubles it is the same.
The result depends on particular case. Same numbers can be exactly converted. Not all. You should keep in mind that.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Question: How to check whether there are new text files in a specified folder and what is the date of creation of the text file.....?

Answer: you can work with the lastmodified timestamp

```
import java.io.*;
public class T {
  public static void main(String[] args) {
    File f = new File("test.txt");
    long l = f.lastModified();
    System.out.println(l);
  }
}
```

- **Question:** How can generate absolute unique number without using complicated math functions? Not pseudo random from Java API... I do not need it for high security; I need to have a unique Id for database connection...

Answer: Combine any of the following:
- IP address
- process id
- random number (from Java API)
- Date(month,day,year)
- Time(hour,minute,second, millisecond)

and then put them together like this:
IPIPIPMonthDayYearHourSecond...Id..RND

- **Advice:** If you are C/C++ programmer be careful with shortcut assignment operator "+=" in Java when you work with Strings!

Why: The shortcut assignment operator += when used with Strings may confuse C and C++ programmers at first. Recall that a += b is equivalent to a = a + b. Let's look at two code samples written in C++ and the Java programming language:

```
//C++ code
string* s1 = new string("hello");
string* s2 = s1;
(*s1) += " world";
cout<<*s1<<endl<<*s2<<endl;
return 0;
//s1 = s2 = "hello world"

//Java programming language code
String s1 = "hello";
String s2 = s1;
s1 += " world";
System.out.println(s1 + "\n" + s2);
//s1 = "hello world" and s2 = "hello"
```

In the C++ example, the strings s1 and s2 print the same result because they both point to the same address. In the Java programming language, Strings can't be modified, so the + operator must create a new String when "world" is appended to s1.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

source: http://java.sun.com/docs/books/tutorial/java/data/stringsAndJavac.html

- **Question:** Why the String does not have a capacity method?

  Answer: The String class doesn't have a capacity method, because a string cannot be changed.

- **Question:** If we write integer or double through Java in a file then the bit pattern for each is reverse than any other language

  (like C/ Delphi writes an integer as qwer but java writes as rewq).

  How can we convert java pattern to other pattern for integer and float/double?

  Answer: If you have an int in the reverse order, you can reverse them manually instead:

```
public int reverse(int i) {

   int result =
   ((i&0xff000000)>>>24)+((i&0x00ff0000)>>>8)+((i&0x0000ff00)<<8)+
   +((i&0x000000ff)<<24);
   return result;
}
```

  and also use this int to convert to a float.

  The new IO in JDK 1.4 handles this situation. ByteBuffer uses instances of java.nio.ByteOrder to determine its byte ordering. The static constants ByteOrder.BIG_ENDIAN and ByteOrder.LITTLE_ENDIAN do exactly what you would expect.

  Essentially, if you talk to another Java program, leave the byte ordering alone and it will work. If you talk to a well-behaved socket application in any language, you should also leave the byte ordering alone. You fiddle with byte ordering in only two instances: when you talk to a poorly-behaved network application that does not respect network byte ordering, or when you deal with binary data files created on a little-endian machine.

  Tip based on article:

  http://www.javaworld.com/javaworld/jw-09-2001/jw-0907-merlin_p.html

- **Question:** I want to get the java.lang.String to read a stream of Chinese characters.

  But I am getting only rubbish characters.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

How can I make my program language independent?

Answer:

You need to have 3 elements:

1. A font.properties file
2. Fonts
3. Swing/AWT components

Add your new font (Chinese) to font.properties file.

Remember that AWT components are limited to displaying characters defined in the host's default code page. For example on an English OS, it is code page 1252.

Until you change a default code page you can not display text properly in a Frame title, a TextArea, or a TextField. If you change it the rest of the system will not be possible to use :-(

The solution is to use only Swing components.

Java 2 allows you to load any font and you will not be limited by fonts that are defined in font.properties file.

- **Question:** I placed the current date and time into String data type. But my table column is in datetime data type.

How do I convert?

Answer: SimpleDateFormat is a concrete class for formatting and parsing dates in a locale-sensitive manner. It allows for

formatting (date -> text), parsing (text -> date), and normalization.

SimpleDateFormat allows you to start by choosing any user-defined patterns for date-time formatting. However, you are encouraged to create a date-time formatter with either getTimeInstance, getDateInstance, or getDateTimeInstance in DateFormat. Each of these class methods can return a date/time formatter initialized with a default format pattern. You may modify the format pattern using the applyPattern methods as desired

```
SimpleTimeZone pdt = new SimpleTimeZone(-8 * 60 * 60 * 1000, "PST");
pdt.setStartRule(DateFields.APRIL, 1, DateFields.SUNDAY, 2*60*60*1000);
pdt.setEndRule(DateFields.OCTOBER, -1, DateFields.SUNDAY, 2*60*60*1000);

// Format the current time.
SimpleDateFormat formatter
          = new SimpleDateFormat ("yyyy.MM.dd G 'at' hh:mm:ss a zzz");

Date currentTime_1 = new Date();
String dateString = formatter.format(currentTime_1);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
// Parse the previous string back into a Date.
ParsePosition pos = new ParsePosition(0);
Date currentTime_2 = formatter.parse(dateString, pos);
```

Tip: What is important to remember when you use SimpleDateFormat?

When parsing a date string using the abbreviated year pattern ("y" or "yy"), SimpleDateFormat must interpret the abbreviated year relative to some century. It does this by adjusting dates to be within 80 years before and 20 years after the time the SimpleDateFormat instance is created. For example, using a pattern of "MM/dd/yy" and a SimpleDateFormat instance created on Jan 1, 1997, the string "01/11/12" would be interpreted as Jan 11, 2012 while the string "05/04/64" would be interpreted as May 4, 1964. During parsing, only strings consisting of exactly two digits, as defined by Character.isDigit(char), will be parsed into the default century.

Any other numeric string, such as a one digit string, a three or more digit string, or a two digit string that isn't all digits (for example, "-1"), is interpreted literally.

So "01/02/3" or "01/02/003" are parsed, using the same pattern, as Jan 2, 3 AD. Likewise, "01/02/-3" is parsed as Jan 2, 4 BC.

If the year pattern has more than two 'y' characters, the year is interpreted literally, regardless of the number of digits. So using the pattern "MM/dd/yyyy", "01/11/12" parses to Jan 11, 12 A.D.

For time zones that have no names, use strings GMT+hours:minutes or GMT-hours:minutes.

The calendar defines what is the first day of the week, the first week of the year, whether hours are zero based or not (0 vs 12 or 24), and the time zone. There is one common decimal format to handle all the numbers; the digit count is handled programmatically according to the pattern.

- **Question:** I have two String arrays: array A and array B. Now I want to get the elements of array A that are not present in array B and vice versa. Can I do that without loop?

  Answer: You can do that with java.util.AbstractSet class.

```
AbstractSet setA = new AbstractSet();
setA.addAll(Arrays.asList(arrayA));
AbstractSet setB = new AbstractSet();
setB.addAll(Arrays.asList(arrayB));
setA.removeAll(setB);
String[] cleanedA = setA.toArray(new String[] {});
```

- **Question:** When do I use String and when StringBuffer?

  Answer: The Java platform provides two classes: String and StringBuffer that store and manipulate strings-character data consisting of more than one character. The

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

String class provides for strings whose value will not change. For example, if you write a method that requires string data and the method is not going to modify the string in any way, pass a String object into the method. The StringBuffer class provides for strings that will be modified; you use string buffers when you know that the value of the character data will change. You typically use string buffers for constructing character data dynamically: for example, when reading text data from a file. Because strings are constants, they are more efficient to use than are string buffers and can be shared. So it's important to use strings when you can.

Source: http://java.sun.com/docs/books/tutorial/java/data/whytwo.html

- **Question:** Why do I get different results here:

```
System.out.println(2 + 2 + "=result");
System.out.println("result=" + 2 + 2);
```

In the first case I get:
4=result
and in the second one:
result=22

Answer: Because Java uses "left to right" principle.
In the first case Java sums up two integers: 2+2=4 and then converts it to a string:
4=result
In the second case Java takes string "result" then converts 2 to a string, concatenates them together.
Then this process repeats with next number... Finally we get: result=22

- **Question:** I need to detect whether a user enters a capitalized letter. I'm using the following code so far:

```
// The keyboard
BufferedReader br =
new BufferedReader(new InputStreamReader(System.in));

String input = null;
do{
  try {
    input = br.readLine();
  } catch (IOException ioe) {
  System.out.println("IO error");
  System.exit(1);
}

//check input for uppercase characters (ignore the
//fact that it might be numbers, assume valid input)
...
```

I have searched the API, in particular, the string function for a way to do this but no luck.

Answer: Try this:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
if(input.equals(input.toLowerCase)) {
   // no upper case letters
   // (untested code)
}
```

Additionally, if you need to test individual characters, char comparisons are perfectly fine.

```
char blah = 'a';
if(blah >= 'A' && blah <= 'Z'){} // if the char is uppercase then...
```

--
Jeff

P.S If you want that code works for all languages, not only English use please do it like this:

```
if (Character.isUpperCase(blah)) { ... }
```

--
Dr. Harald M. Muller
Senior Consultant - OO, Java, SW-Architecture
Cortex Brainware, WWW: http://www.cortex-brainware.de

- **Question:** An idea for validating the phone number field on a form..,

Does anyone have an idea for validating the phone number field on a form? I am looking for something that will basically check this input mask

*111*111*1111*

Where the 1's are number's and the *'s are either - . spaces, or any other character like (). Please advise.

Answer 1: You could use a regular expression package. For example, Jakarta ORO: http://jakarta.apache.org/oro/

Answer 2: i'm thinking regular expressions. See:
http://www.cacas.org/java/gnu/regexp/
http://www.crocodile.org/~sts/Rex/

- **Question:** Could someone show me a basic File I/O example? I just can't figure out streams. I'm willing to accept basic mockery in exchange...

Could someone show me a basic File I/O example? I just can't figure out streams. I'm willing to accept basic mockery in exchange...

Answer:

```
import java.io.*;

public class FileIO {

   public static void main(String[] args) throws Exception {
     if(args.length!=1){
        System.out.println("Invalid parameters!!!");
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
        System.exit(0);
      }
      File fl = new File(args[0]);
      FileReader fileReader = new FileReader(fl);
      BufferedReader bufferedReader = new BufferedReader(fileReader);
      String currentLine;
      while( (currentLine = bufferedReader.readLine()) != null ){
        System.out.println(currentLine);
      }
    }
}
```

- **Question:** Does anybody know a convenient way to pause the dos program execution until user hits enter? In C I used getc. Does Java have an equivalent of "cin>>"?

  Answer:

```
try {
   System.in.read()
} catch (Exception e) {
}
```

  Have fun!
  --
  Bary

- **Question:** I've got a (simple) menu on a new application and am trying to put in the works behind the cut, copy & paste menu options - does anyone know how I can do this - what's the code or can you point me in the right direction?

  Answer: Look at java.awt.datatransfer package. It contains much of the tools necessary to implement cut. copy, paste.

  Can anyone please explain clearly how BufferedReader works and how to use it to get input from a keyboard?

- **Question:** Can anyone please explain clearly how BufferedReader works and how to use it to get input from a keyboard?

  Answer: BufferedReader is a filter reader class.
  That is, it wraps another reader and reading from it is like reading from the reader it wraps, except that it changes something. In the case of BufferedReader, it reads in large chunks and then you can retrieve its data in smaller bits. To use it to read from System.in, you first need a reader to wrap. You can bridge from an input stream (which System.in is) to a reader by using an InputStreamReader.
  Then wrap that in a BufferedReader as follows:

```
BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
```

  Now you can call methods of BufferedReader to read from standard input. Generally, you create a BufferedReader to be able to call the readLine() method. That isn't

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

BufferedReader's main intended use -- the main intended use is performance -- but you don't generally care too awfully much about performance of reads from the console. So call readLine to get a line of input, which will be null on end of stream (user presses Ctrl-D on UNIX or a file was redirected in and is done).
Answered by Chris Smith

- **Question:** How do I encode the value of a variable of type long (or int) into bytes? And how do I restore the original value of the long (or int) variable back

  How do I encode the value of a variable of type long (or int) into bytes such that the number of bytes used will always be the same, say 4 bytes?

  Answer:

```
int in; ...
byte b1 = (byte)(in & 0xff);
byte b2 = (byte)((in >> 8) & 0xff);
byte b3 = (byte)((in >> 16) & 0xff);
byte b4 = (byte)(in >>> 24);
```

  : How do I restore the original value of the long (or int) variable back
  : from the bytes that i have just created then?

  Answer:

```
int in = (b1 & 0xff) | ((b2 << 8) & 0xff00) |
((b3 << 24) >>> 8) | (b4 << 24);
```

  by Tim Tyler

- **Question:** I would like to know if there was a method that replaces a substring (in a string) by another one.

  For ex: in the string: "toto=yes, tata=no", I want to replace "yes" by "no", and "no" by "N/A",

  Answer: There isn't a method to do it, since Strings are immutable, so you can't directly replace elements. (If you're doing a lot of string manipulation, shift the string to a StringBuffer.)
  However, it's not hard to create a new string that performs the replacement, along the lines of:

```
int p = source.indexOf (target);
String result = source.substring (0, p) +
replacement + source.substring (p + target.length ())
```

  In your example, beware of the order of replacement, since the second no will replace the first one.

  Matt Humphrey

- **Question:** I have a String containing multiple logical lines separated with "\r\n". I want to tokenize that string to individual lines.
  However, StringTokenizer doesn't work the way I would like it to work with "empty" lines.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Example:
String str = "first line\r\n" +
"\r\n" +
"third line\r\n");

I would like StringTokenizer to tokenize the string to the following
three strings:
"first line"
""
"third line".

I have set "\r\n" to delim, but StringTokenizer.nextToken doesn't seem to return the second, empty line.

I also tried setting the returnDelims in StringTokenizer constructor to false, but then nextToken returns the "\r" and "\n" as separate strings. It should return "", in order to make the rest of my code to work well.

How should I solve this problem?

Answer: There are several ways to get around this problem with utilities that I have written.

The easiest way is to use a split function.

String[] stuff = StringHelper.split("first\r\n\r\nthird", "\r\n");

stuff will be a string array with 3 elements, including the empty line
in the middle. You can get StringHelper from:

http://ostermiller.org/utils/StringHelper.html

The second way is to use a StringTokenizer that returns empty tokens.
I have written a StringTokenizer that does such at:

http://ostermiller.org/utils/StringTokenizer.html

The only problem that you would run into is that StringTokenizer is a character tokenizer. So you would get extra empty tokens in between the \r and the \n.
--
Stephen

Ł **Question:** I'm looking for a solution for parsing a String object. I need to parse a math expression and calculate it.

At first, I thought of StringTokenizer, but when I looked through the API document, multiple delimiters might not be allowed.
What is the best way to do this operation?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: multiple delimiters _are_ allowed.
StringTokenizer st = new StringTokenizer(your_string, "+-/* ")
and so on.

- **Question:** Are there any public domain matrix/linear algebra java packages out there?

I'm mainly interested in solving systems of equations and finding matrix inverses.

Answer: http://math.nist.gov/javanumerics/
The JavaNumerics page provides a focal point for information on numerical computing in Java.

- **Question:** I am using the sinusoidal wave calculated by feeding a in function sin(a) values from zero to pi (3.142).I am wondering how is the Math.sin implemented in respect to effective calculating in practice:

Should I prepare a table with pre-calculated sine-values, or does java already have that table prepared for me?
If I 'pre-calculate' that table once, it'll make about 1024 sin-calculations plus all the lookups from the table (r, g, b + a values (4 * 255 values * 4 transitions)).
Otherwise it's just the Math.sin-function call for the aforementioned RGB-values.

Answer: Most likely the JVM utilizes the floating point unit (FPU) of your microprocessor. The FPU is hard to beat these days in software. You could try a Taylor sequence, and Chebyshev approximations, but I doubt that you will be faster. On Intel, it is a hardware instruction. It is implemented internally as a polynomial approximation. You are best to treat it as a black box.
Profile it.
If speed is a problem, pre-calculating a table is an option. It's the old trade-off memory vs. performance.
--
Thomas Weidenfeller

- **Question:** Does anyone know how to write multi-line string in Java?

Answer: Something like this:

```
String a =
    "This is a
         multiline string.";
```

It is really pain to print HTML or XML from the Java program. Perl offer something like this:

```
print <<END_HTML;
  <html>
     <body>
       <h1>this is html.</h1>
     </body>
</html>
END_HTML;
```

Answer: Try this:
Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

String a = "This is a\nmultiline string"

A "\n" stands for a line feed. Take a look at the Java language specification (downloadable on Sun's site), it has a section about strings.

Answer 2: You mean like this?

```
String a = "<html>" +
           " <body>" +
           " <h1>this is html.</h1>" +
           " </body>" +
           "</html>";
```

- **Question:** I cannot find the method to convert a binary number to an int. I used Integer.toBinaryString to get a decimal to binary but I don't know how to convert it back.

  Answer: Try using Integer.parseInt(String s, int radix) with radix = 2 that should do your job.

- **Question:** How do I launch a native Document by its Associated MIME Type? For example, I would like to ask the 'operating system' what application is associated with .DOC and then launch it.

  Answer: On WinNt,

```
String docName = "c:\\someyourdir\\nameofdoc.doc";
Runtime.getRuntime().exec("cmd.exe /c "+docName);
```

- **Question:** How do I indicate Unicode characters that cannot be represented in ASCII, such as •?

  Answer: from "Java Tutorial (http://java.sun.com/docs/books/tutorial/i18n/text/convertintro.html) "To indicate Unicode characters that cannot be represented in ASCII, such as o, we used the \uXXXX escape sequence. Each X in the escape sequence is a hexadecimal digit. The following example shows how to indicate the o character with an escape sequence:

```
String str = "\u00F6";
char c = '\u00F6';
Character letter = new Character ('\u00F6'); "
```

- **Question:** When I tried to read one string representing boolean value and convert it into boolean it didn't work. Finally I found that Java API has a bug!

  I wrote the program that uses reading ini file settings for initialization. All settings in a file are strings. I am converting them to appropriate type during reading. When I tried to read one string representing boolean value and convert it into boolean it didn't work. Finally I found that Java API has a bug:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

boolean x = true;
getBoolean(x);
will show false!!!!

Why Java has method that doesn't work? Is it bug in Java or I am stupid?

Answer: neither statement is true! It is not a bug and you are Ok! Just please read more carefully JavaDoc next time.

It is written there for getBoolean ():

"Returns is true if and only if the system property named by the argument exists and is equal to the string "true".

(Beginning with Java 1.0.2, the test of this string is case insensitive.)

A system property is accessible through getProperty, a method defined by the System class."

So you didn't use this method properly...
Use instead:

public static Boolean valueOf(String s)

This method returns the boolean value represented by the specified String. A new Boolean object is constructed. This Boolean contains the value true if the string argument is not null and is equal, ignoring case, to the string "true".

example:
boolean x= true;
(Boolean.valueOf(x)).booleanValue()
gives you proper boolean (not Boolean!) value

- **Question:** Is there any Java API allowing creating easily PDF files (Adobe Acrobat type) including images?

Answer: No, just text
Etymon™ PJ is a developer toolkit for parsing, modifying, and creating PDF documents.
http://www.etymon.com/pj/index.html

- **Question:** I'm working on a java project and looking for a better API that can generate PDF, work with Excel, Word documents... Where can I find it?

Answer:

iTest
This library contains classes that generate documents in the Portable Document Format(PDF) and/or HTML

XML->PDF
FOP is the world's first print formatter driven by XSL formatting objects. It is a Java

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

application that reads a formatting object tree and then turns it into a PDF document. The formatting object tree, can be in the form of an XML document (output by an XSLT engine like XT or Xalan) or can be passed in memory as a DOM Document or (in the case of XT) SAX events.

JPageLayout from Sitraka
JClass PageLayout provides sophisticated and easy-to-use APIs for adding text, images, and tables to any document. Output directly to the Java AWT Printer, Acrobat PDF, HTML, PostScript Level 2, or PCL 5. Customize almost every aspect of your report styles and print output for professional results.

- **Question:** How can I extract text from PDF file?

  Answer: I have used the Acrobat Viewer Bean:

  (http://www.adobe.com/products/acrviewer/main.html) to extract text from PDFs.

  This bean is quite buggy but it was OK for this task. One drawback is that it depends on AWT even if you don't do anything GUIish with it, so if you want to use it on a Server with no X-Windows running you'll have to install the Pure Java AWT to get things running.
  --
  Patrick

- **Question:** I'm looking for a rich text editor that I can embed within a web page, and allow users to enter rich text that I can in turn store as HTML.

  I've seen similar applets through web based e-mail clients. I'd appreciate it if someone could point me in the right direction!

  Answer:
  Try Swing, by Robinson, Manning Publication. You could probably adjust the code to fit into the applet style.
  It is here

  http://javafaq.nu/java/free-swing-book/free-swing-book-chapter20.shtml

  John

- **Question:** I want to have a while loop execute for a maximum of n seconds or until it receives something in an input stream. How would I do this?

- **Answer:** I think you could do it this way:

```
**********************************************

InputStream Input=null;

int n=10; /*Number of seconds to wait*/

/*initialize you input stream*/

...

/*Now start you while loop*/
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
long lStart = (new Date()).getTime();
long lCurrent = (new Date()).getTime();

while((lCurrent-lStart < n*1000) && (Input.available()==0)){
   Thread.currentThread.sleep(100); /* This will give JVM time to other threads */
   lCurrent = (new Date()).getTime();
}
**********************************************
```

You could simply count number of steps inside the loop keeping in mind that each step takes 100ms but I think using dates would be a bit more precise method.
Alex Shlega

Question: How do I parse it to date and make a date difference? How many days in between the String from and to?

I am always confusing with date and time.

I have 2 String, from and to
String from = "01.01.2001" //dd.MM.yyyy
String to = "01.03.2001" //dd.MM.yyyy

How do I parse it to date and make a date difference?? How many days in between the String from and to?

Answer:

```
import java.text.SimpleDateFormat;
import java.util.Date;
public class Tmp {
  public static void main( String argv[] ) throws Exception {
    long DAY = 24L * 60L * 60L * 1000L;
    SimpleDateFormat df = new SimpleDateFormat( "MM.dd.yyyy" );
    Date d1 = df.parse( "01.01.2001" );
    Date d2 = df.parse( "01.03.2001" );
    System.out.println( "The number days between:" );
    System.out.println( d1 );
    System.out.println( "and:" );
    System.out.println( d2 );
    System.out.println( "is: " + (( d2.getTime() - d1.getTime() ) / DAY ));
  }
}
```

But the calculation of the difference in times may not be suitable due to timezone issues and such. I believe there must be a better way?

Michael B. Allen

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** I am looking for some code to compare current date and a date entered by user. I want to return years between dates. I can't find a method to do this with.

  Answer: try GregorianCalendar class which is found in java.util.*

```
Calendar fdate= new GregorianCalendar(1999,1,25);
Calendar ldate= new GregorianCalendar(2000,1,25);

then use get methods and compare
fyear= fdate.get(Calendar.YEAR);
lyear= ldate.get(Calendar.YEAR);
```

- **Question:** If I have a string with a hex value (e.g. "7FFFFF"), is there a way to obtain an int value from this (e.g. 8388607)?

  Answer: See Integer.parseInt (String s, int radix). The radix for a hexadecimal number is 16.

- **Question:** How do I convert this double to the following String:

```
double d = 19.969332079021637;
String s = "19.97";
```

  Answer:
```
double d = 19.969332079021637;
DecimalFormat df = new DecimalFormat("##,###.##");
String s = df.format(d);

or

Use NumberFormat in the java.text package
...
double d = 19.969332079021637
NumberFormat format = NumberFormat.getInstance();
format.setMaximumFractionDigits(2);
format.setMinimumFractionDigits(2);
String s = format.format(d); // s="19.97"
```

- **Question:** How do I convert a String to an integer. For example: String strSEQ = "SEQ7";

  How do I pull "7" from the string and convert it to an integer?

  Answer: I'd do something like this:

```
StringBuffer digits = new StringBuffer();
char c;
for (int i=0; i<strSEQ.length(); i++) {
   c = strSEQ.charAt(i);
   if (Character.isDigit(c))
      digits.append(c);

}
int intValue = Integer.parseInt(digits.toString());
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

--
Michiel

- **Question:** How can I display other languages such as Chinese, French, and Japanese etc. inside applet or frame?

  Answer: You have to install the appropriate font, and add it to a font.properties file. Further information can be found here:

  http://forum.java.sun.com/read/16805306/q_CUP8NP-1rgAAYsA#LR
  http://java.sun.com/products/jdk/1.1/docs/guide/intl/index.html
  http://developer.java.sun.com/developer/qow/archive/65/index.html

- **Question:** I need to determine the current year, at runtime, in the form of a four digit number, preferably in "int" form. How can I do this?

  Answer: Hey, what makes you think that the year is a four digit number? It won't always be, you know. That's the kind of sloppy thinking that got us into that whole Y2K mess, and we all know what a huge issue THAT turned out to be!

  If you write your program assuming that the year is 4 digits, it will cease to function properly in less that 8000 years! Try not to have such a "short term" viewpoint.

  Here's the deprecated solution:

  int year = new java.util.Date().getYear() + 1900;

  This was deprecated because it is too convenient! We programmers often get paid per line of code, you know.
  Actually it was deprecated because it didn't have an implicit way to deal with time zones and it contains that silly Y2K problem.

  The SUPERIOR way to do is as follows:

  java.util.Calendar cal = Calendar.getInstance();
  int year = cal.get(Calendar.YEAR);

  --
  Marshall

- **Question:** I have a string like 23,N/A,,N/A,87, then I used the StringTokenizer t = new StringTokenizer(s, ",");

  to get the token. I seems I only got 4 tokens back, not 5. How doesStringTokenizer treat the ,, in this case? I had tried String.equals("") or String.equals(null) to pick the EMPTY string between ,, , it did not work.
  Anyone can give a clue?

  Answer: There is another constructor, with a third parameter. Using that one, you can get the` boundary markers, such as the comma, returned to you as well. That would allow you to handle your empty strings.
  --
  Michele.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

- **Question:** How to know in Java if a given font has or not a fixed width?

  For example to list all the "fixed width" fonts that are available on the user's platform.

  Answer: I'm not sure whether there's an easier solution... if there is none, you might try to get the FontMetrics objects from a specific Font. Then iterate over every character and call charWidth(char ch) on each. It should return the same value for each character if it's monospaced, different values for each one otherwise.

  This might be a little "hacky", but if you really need an idea...

  Karsten

- **Question:** I want to create a Date object by just specifying the Year, Month and Day.

  e.g. the user can enter 1998 as Year, 11 as Month of Nov, 15 as the day.
  With these three information, how can I create a Date object (or similar object that stores the data)?

  Answer: The correct way to do this is something like this:

  ```
  Calendar cal = Calendar.getInstance();
  cal.set(1998, Calendar.December, 15);
  Date date = cal.getTime();
  ```

- **Question:** Let's say I have a text string with the text: Hello! Then how do I add quotes to it so it will be: "Hello!"

  Answer:
  answer is String s = "Hello!"

  ```
  "\"" + s + "\""
  ```

  and if the result is to become the new value of s,

  ```
  s = "\"" + s + "\""
  ```

- **Question:** I want to send Image through network. I made OutputStream to send Image...

  ```
  OutputStream out = socket.getOutputStream();
  ```
  but I can't send Image directly.
  How can I send Image?

  Answer: Unfortunately, the java.awt.Image class is abstract and is not serializable, otherwise this would be easy.

  From the Java docs: "The abstract class Image is the superclass of all classes that represent graphical images. The image must be obtained in a platform-specific manner."

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Consider using the Image.getSource method, getting the producer, getting the bytes, then send [width][height][bytes...].
--
Mark Watson, Java consulting, Open Source and Content: www.markwatson.com
Commercial software products: www.knowledgebooks.com

- **Question:** I want to get the code behind an HTML page.

Answer:

```
URL url = new URL("http://www.sun.com");
BufferedReader reader;
reader = new BufferedReader(new InputStreamReader(url.openStream()));
String str;
while ((str = reader.readLine())!=null){
  // process str
}
```

- **Question:** I am wondering if JDK supports to open a file in the exclusive mode.

Answer: No, file locking is not supported in current VMs. You can implement lockfiles, however, using the File.createNewFile() method. Since JDK 1.3: "Atomically creates a new, empty file named by this abstract pathname if and only if a file with this name does not yet exist. The check for the ' existence of the file and the creation of the file if it does not exist are a single operation that is atomic with respect to all other filesystem activities that might affect the file. This method, in combination with the deleteOnExit() method, can therefore serve as the basis for a simple but reliable cooperative file-locking protocol."

- **Question:** How do I write an image to stdout? This is for a cgi program that will return an image to a browser.

For example, if a user click on a link to http://somewhere.edu/cgi/getpix.cgi the browser will receive image data from the cgi program and will display the image.

Answer: Send "Content-type: image/gif" - and then send the data.

Alternatively, send an HTML page, with a pointer to the (possibly-freshly-written) image on the server.
--
Tim Tyler Try my latest game - it rockz - http://rockz.co.uk/

- **Question:** Does anyone know how I can get around the fact that a Message obj (from the JavaMail API) is not serializable?

I would like to save a message obj to the hd, then read the file (is 'file' the wrong word here?) back to my e-mail client and read it as a Message obj.

Answer: The obvious answer is: implement serializable yourself in a subclass! But! That won't work. The serialization mechanism will throw an exception at runtime. Even if it didn't, the class may be nonserializable for a good reason, for example it might contain a socket, which can't be reinitialized from a stream.

What you need is a subclass that implements the Externalizable interface, and then

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

your own code that saves as serializable objects those attributes of the object that you need to reconstitute it later. I'm not familiar with this particular object, but I imagine it has several get() methods that return serializable objects like Strings or ints. What you have to do is recover those objects and put them into the output stream yourself, and reverse the process on unserialization. The fields that are not serializable you ignore.
You have to provide a no-argument constructor that reconstructs anything you aren't going to serialize (for example, opening a new socket). Read about the Externalizable interface for more details.
--
Frank LaRosa

- **Question:** Is it possible to send Vector class in socket communication. If it is, what function can make it.

Answer: Make sure the objects contained in the Vector are Serializable. Look at java.io.ObjectOutputStream for serializing the Vector by wrapping the output stream from socket with the ObjectOutputStream...

- **Question:** I wrote a little Java Tool that imports a 10 MB Text file. My Problem is that this text file has no line break that means the whole file contains one record, which are all records put to one together.

With the readLine () method it takes more than an hour to complete my instruction.

How can I accelerate this process, I only need to insert a break line after each "record" (after x characters -> "\n").

Answer: If you will tell us what you are trying to accomplish, we may be able to offer a suggestion.

If you are trying to reformat the file to have line breaks, you will have to read the entire file once and write out a new file with the line breaks. It seems obvious that the original file was meant to be random-accessed and has fixed-length records, in which case it is just fine as it is -- if it is read appropriately.
--
Paul Lutus www.arachnoid.com

- **Question:** How can you tell if an integer is odd or even?

I know an even number is divisible by 2 but I'm thinking then how can I detect if a resulting number after dividing 2 ints has a remainder?

Answer: Basically the method is simple, if a variable contains an odd number I want to return the value 'zero', and on the other hand if the variable contains an even number I want to return the value 'one'.

Check out the % (modulus) operator, it computes the remainder.

```
public int modMethod(int div){
    if(div%2==0){
        return 1;
   } else{
       return 0;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
  }
}
```

- **Question:** I'm looking for an algorithm that would compress a large integer down to a not-that-long string. Actually, it's not an integer, but a series of integers that could have hundreds of digits in total.

  Answer: Use classes ZipInputStream and ZipOutputStream. They are meant for exactly this purpose. And they meet the requirement of portability between languages (because ZIP compression is ubiquitous).
  --
  Paul

- **Question:** How can I round a number to specified precision?

  I have a double field that I would like to round to 2 places of precision, however, it seems like the documentation on the round function only rounds to closest integers. So that I would not be able say .3658585859 = .37 as I would like.

  Answer: can you scale the number up and then down again when you are finished?
  e.g. 0.3658585859 * 100 = 36.58585859
  round(36.58585859) = 37
  37 / 100 = 0.37

- **Question:** I understand that bitwise operations change the 0/1 bits of a number. Question is why?

  I suppose it's interesting that you can manipulate numbers this way, but I can't think of a practical use for doing that.

  Can anyone help me understand when are bitwise operations used and why you would use them?

  Answer: Bitwise manipulation is often used where memory consumption is critical, and a piece of information may be encoded in less that one byte, for instance. In communication software and protocols, information may be interpreted as a stream of bits where the information is encoded at the bit-level, and you use bitwise manipulation to extract the pieces of information encoded in the bytes. There are other situations where bitwise manipulation is used, as well.
  by Greger Ohlson

- **Question:** Why cannot I cast from double to java.lang.Object?

  I'm trying to build a vector, however, one of the objects that I'm passing to the vector is of type double. How do I cast the double as an object so that I may insert the value into a vector? Does this make sense? Here is the following snippet of code I was trying to use:

  myVector.add (1, (Object)myDouble);

  Of course when I try to compile I get the following message:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Invalid cast from double to java.lang.Object

Could someone please explain why? I realize that Object is the mother of all objects and therefore ANY reference data type "is an" Object. So therefore I shouldn't have to cast the double, right? Help, I'm a beginner!

Answer: A double is not a reference type, but a primitive one. Hence, it doesn't inherit from Object (or anything else, for that matter). To put primitives (byte, short, int, long, float, double, boolean, char) into something that requires an Object, use Java's wrapper classes.

The wrapper classes are Double, Integer, Long, Boolean, etc., and are basically an object "wrapped" around a primitive type. You make a Double object by:
Double d = new Double (myDouble);

and to get the actual value back,

double z = d.doubleValue();

It works the same way for all the rest of the primitive/wrapper pairs.
by Trevor Hill

- **Question:** is there a mod (x, y) function that returns the remainder when x is divided by y? Something equivalent to fmod(x,y) in C?

  Answer: a = x%y;

- **Question:** I'm having trouble figuring out how to convert characters to their ASCII value in java. Is there a class like NumberFormat that will do it?

  Answer: I can't see any problem here:

  char ch = 'A'; // character 'A'
  int i = (int)ch; // ASCII value for 'A' (=>65)
  Yes. And just be aware that ASCII only runs from 0 through 127. Anything higher needs to be addressed differently, since Java is using Unicode values.

- **Question:** How to do "Press any key to continue"? I want to do it at Console.

  Answer: // ReadConsole.java

```
import java.io.*;

public class ReadConsole {

   public static void main(String args[]) throws IOException {
      System.out.print("Press Enter to continue: ");
      System.in.read();
   }
}
```

You cannot have "press any key" from the console, for various system-dependent reasons. You need to press Enter.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

--
Paul Lutus www.arachnoid.com

- **Question:** Just wondering how people generally convert BufferedOutputStream into a BufferedInputStream to be read from....

  This seems really stupid, but I can't find a way to do it in the API....

  Answer: if you want to just take what's coming in and send it out then do something like this

```
BufferedInputStream in = new BufferedInputStream(some inputstream);
BufferedOutputStream out = new BufferedOutputStream(some outputstream);

int i = 0;

while((i = in.read()) != -1){
out.write(i);
}
```

  of course you will have to handle exceptions, but that should be the general way to do it.
  --
  michael

- **Question:** I have heard that String concatenation operator + affects performance of program if it used much. Is it true?

- **Answer:** Yes, it affects your program performance if you do a lot of "+" operations with strings:
  A new StringBuffer must be created, then two arguments are added to it with append(), and the final result must be converted back with a toString().
  Your time and space is wasted...
  In case if you are appending more than one String, try to use a StringBuffer directly.

21. Threads

- **Question:** Can anyone please explain the difference between the two types: green threads and native threads?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: Green threads is thread mechanism implemented in JVM itself. It is blind and can run on any OS, so actually all threads are run in one native thread and scheduling is up to JVM. This is disadvantageously for SMP systems, since only one processor can serve Java application.

Native threads is a mechanism based on OS threading mechanism. This allows to use features of hardware and OS. For example, there is IBM's JDK for AIX that supports native threads. The performance of applications can be highly improved by this.

- **Question:** I use myThread.isAlive() method to test that thread is still alive. But sometime even if it indicates that some thread is still running I am getting errors when try to use that thread. What is wrong?

Answer: the method is Alive makes you sure that thread actually "was" not "is" alive at the moment when you used it. I think it could be better if SUN rename it to wasAlive() and did not confuse people. This method just checks once when you use it and does not keep eye on what happens after.

Probably the thread will exit right after that.

The method activeCount() does the same: it counts all running threads. This information becomes obsolete immediately next moment!
enumerate() method is another example.

All those methods are useful only for statistic collection, nothing else. Avoid using them for other purposes!
To know when the thread exits use join() method.
--
AP (J.A.)

- **Question:** Why wait(), sleep(), notify(), notifyAll() must be inside a synchronized block, otherwise it throws java.lang.IllegalMonitorStateException: current thread not owner.

Answer: Wait/notify unlocks the monitor and it is not synchronized another thread can try to access it and start a race condition.

- **Question:** The thread constructor since 1.4.1 allows the thread stack size to be specified. What can I do with that?

Answer: Yes, in 1.4.1 we have the new constructor:
public Thread(ThreadGroup group, Runnable target, String name,long stackSize)

Setting a larger value for stackSize may allow more levels of recursion before a StackOverflowError occurs. Setting a smaller value may also allow a larger number of concurrent threads to exist before an OutOfMemoryError or another internal error is thrown. However, the VM may ignore the stackSize setting, and the actual effect of the setting is VM dependent. Therefore, any tuning based on this parameter may

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

have to be redone for each Java Runtime Environment (JRE) on which the application will run...

More info here:
http://java.sun.com/j2se/1.4.2/docs/api/java/lang/Thread.html#Thread(java.lang.ThreadGroup,%20java.lang.Runnable,%20java.lang.String,%20long)

The tip is based on SUN's "Version 1.4 Core Libraries" description

- **Question:** I read many times that Thread.stop is deprecated and it is unsafe to use it. But never got good explanation why....

  Answer: Because it is inherently unsafe. Stopping a thread causes it to unlock all the monitors that it has locked. (The monitors are unlocked as the ThreadDeath exception propagates up the stack.) If any of the objects previously protected by these monitors were in an inconsistent state, other threads may now view these objects in an inconsistent state. Such objects are said to be damaged. When threads operate on damaged objects, arbitrary behavior can result. This behavior may be subtle and difficult to detect, or it may be pronounced. Unlike other unchecked exceptions, ThreadDeath kills threads silently; thus, the user has no warning that his program may be corrupted. The corruption can manifest itself at any time after the actual damage occurs, even hours or days in the future.

  http://java.sun.com/products/jdk/1.2/docs/guide/misc/threadPrimitiveDeprecation.html

- **Question:** How do I stop a thread that waits for long periods (e.g., for input)?

  Answer: That's what the Thread.interrupt method is for. The same "state based" signalling mechanism shown above can be used, but the state change (blinker = null, in the previous example) can be followed by a call to Thread.interrupt, to interrupt the wait:

```
public void stop() {
  Thread moribund = waiter;
  waiter = null;
  moribund.interrupt();
}
```

  For this technique to work, it's critical that any method that catches an interrupt exception and is not prepared to deal with it immediately reasserts the exception. We say reasserts rather than rethrows, because it is not always possible to rethrow the exception. If the method that catches the InterruptedException is not declared to throw this (checked) exception, then it should "reinterrupt itself" with the following incantation:

```
Thread.currentThread().interrupt();
```

  This ensures that the Thread will reraise the InterruptedException as soon as it is able.
  http://java.sun.com/products/jdk/1.2/docs/guide/misc/threadPrimitiveDeprecation.h

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

tml

- **Question:** What if a thread doesn't respond to Thread.interrupt?

Answer: In some cases, you can use application specific tricks. For example, if a thread is waiting on a known socket, you can close the socket to cause the thread to return immediately. Unfortunately, there really isn't any technique that works in general. It should be noted that in all situations where a waiting thread doesn't respond to Thread.interrupt, it wouldn't respond to Thread.stop either. Such cases include deliberate denial-of-service attacks, and I/O operations for which thread.stop and thread.interrupt do not work properly.

This question/answer was published based on SUN's article:
http://java.sun.com/products/jdk/1.2/docs/guide/misc/threadPrimitiveDeprecation.html

- **Question:** Why are Thread.suspend and Thread.resume deprecated?

Answer: Thread.suspend is inherently deadlock-prone. If the target thread holds a lock on the monitor protecting a critical system resource when it is suspended, no thread can access this resource until the target thread is resumed. If the thread that would resume the target thread attempts to lock this monitor prior to calling resume, deadlock results. Such deadlocks typically manifest themselves as "frozen" processes.

This question/answer was published based on SUN's article:
http://java.sun.com/products/jdk/1.2/docs/guide/misc/threadPrimitiveDeprecation.html

- **Question:** What should I use instead of Thread.suspend and Thread.resume? Part 1.

Answer: As with Thread.stop, the prudent approach is to have the "target thread" poll a variable indicating the desired state of the thread (active or suspended). When the desired state is suspended, the thread waits using Object.wait. When the thread is resumed, the target thread is notified using Object.notify.

For example, suppose your applet contains the following mousePressed event handler, which toggles the state of a thread called blinker:

```
private boolean threadSuspended;
Public void mousePressed(MouseEvent e) {
  e.consume();

  if (threadSuspended)
     blinker.resume();
  else
    blinker.suspend(); // DEADLOCK-PRONE!

  threadSuspended = !threadSuspended;
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can avoid the use of Thread.suspend and Thread.resume by replacing the event handler above with:

```
public synchronized void mousePressed(MouseEvent e) {
e.consume();

threadSuspended = !threadSuspended;

if (!threadSuspended)
  notify();
}
```

and adding the following code to the "run loop":

```
synchronized(this) {
  while (threadSuspended)
    wait();
}
```

The wait method throws the InterruptedException, so it must be inside a try ... catch clause. It's fine to put it in the same clause as the sleep. The check should follow (rather than precede) the sleep so the window is immediately repainted when the thread is "resumed." The resulting run method follows:

```
public void run() {
  while (true) {
    try {
      Thread.currentThread().sleep(interval);
      synchronized(this) {
        while (threadSuspended)
          wait();
      }
    } catch (InterruptedException e){
    }
    repaint();
  }
}
```

Note that the notify in the mousePressed method and the wait in the run method are inside synchronized blocks. This is required by the language, and ensures that wait and notify are properly serialized. In practical terms, this eliminates race conditions that could cause the "suspended" thread to miss a notify and remain suspended indefinitely.

- **Question:** What should I use instead of Thread.suspend and Thread.resume? Part 2.

While the cost of synchronization in Java is decreasing as the platform matures, it will never be free. A simple trick can be used to remove the synchronization that we've added to each iteration of the "run loop." The synchronized block that was added is replaced by a slightly more complex piece of code that enters a synchronized block only if the thread has actually been suspended:

```
if (threadSuspended) {
  synchronized(this) {
    while (threadSuspended)
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
        wait();
    }
}
```

In the absence of explicit synchronization, threadSuspended must be made volatile to ensure prompt communication of the suspend-request.

The resulting run method is:

```
private boolean volatile threadSuspended;

public void run() {
  while (true) {
    try {
      Thread.currentThread().sleep(interval);
      if (threadSuspended) {
        synchronized(this) {
          while (threadSuspended)
            wait();
        }
      }
    } catch (InterruptedException e){
    }
    repaint();
  }
}
```

this question/answer was published based on SUN's article: http://java.sun.com/products/jdk/1.2/docs/guide/misc/threadPrimitiveDeprecation.html

- **Question:** Can I combine the two techniques to produce a thread that may be safely "stopped" or "suspended"?

Answer: Yes; it's reasonably straightforward. The one subtlety is that the target thread may already be suspended at the time that another thread tries to stop it. If the stop method merely sets the state variable (blinker) to null, the target thread will remain suspended (waiting on the monitor), rather than exiting gracefully as it should. If the applet is restarted, multiple threads could end up waiting on the monitor at the same time, resulting in erratic behavior.

To rectify this situation, the stop method must ensure that the target thread resumes immediately if it is suspended. Once the target thread resumes, it must recognize immediately that it has been stopped, and exit gracefully. Here's how the resulting run and stop methods look:

```
public void run() {
  Thread thisThread = Thread.currentThread();
  while (blinker == thisThread) {
    try {
      thisThread.sleep(interval);
      synchronized(this) {
        while (threadSuspended && blinker==thisThread)
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
            wait();
          }
      } catch (InterruptedException e){
      }
      repaint();
    }
  }

  public synchronized void stop() {
    blinker = null;
    notify();
  }
```

If the stop method calls Thread.interrupt, as described above, it needn't call notify as well, but it still must be synchronized. This ensures that the target thread won't miss an interrupt due to a race condition.

this question/answer was published based on SUN's article: http://java.sun.com/products/jdk/1.2/docs/guide/misc/threadPrimitiveDeprecation.html

- **Question:** What about Thread.destroy?

  Answer: Thread.destroy has never been implemented. If it were implemented, it would be deadlock-prone in the manner of Thread.suspend. (In fact, it is roughly equivalent to Thread.suspend without the possibility of a subsequent Thread.resume.) We are not implementing it at this time, but neither are we deprecating it (forestalling its implementation in future). While it would certainly be deadlock prone, it has been argued that there may be circumstances where a program is willing to risk a deadlock rather than exit outright.

- **Question:** Why is Runtime.runFinalizersOnExit deprecated?

  Answer: Because it is inherently unsafe. It may result in finalizers being called on live objects while other threads are concurrently manipulating those objects, resulting in erratic behavior or deadlock. While this problem could be prevented if the class whose objects are being finalized were coded to "defend against" this call, most programmers do not defend against it. They assume that an object is dead at the time that its finalizer is called.

  Further, the call is not "thread-safe" in the sense that it sets a VM-global flag. This forces every class with a finalizer to defend against the finalization of live objects!

  this question/answer was published based on SUN's article: http://java.sun.com/products/jdk/1.2/docs/guide/misc/threadPrimitiveDeprecation.html

- **Question:** What are Thread Groups useful for? I do not see big reason to have them in Java API...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Answer: From the beginning Thread Groups were added for security reasons. Assumed, that not trusted code from third parties will run in dedicated Thread Group. Of course it can not affect the rest of program. Those threads from external companies were not allowed to start, stop and suspend your threads. Then this model is changed and we have no such protection now.
Now the Thread Groups will let you a little bit easier manage your threads. You can manage many threads simultaneously rather than each separately. For example you can set maximum priority level, call suspend(), interrupt() on all threads in your group. Also giving good names will help you to easily track and debug your program.

- **Question:** I get an error "Green threads not supported" when tried to use -green option on Windows.

  But my application runs just fine...

  Does this mean that by default JVM is using native option?

  Answer: Just Solaris and Linux support the option -green threads. Green threads are independent of native threads and less efficient. The downside is that using green threads means system threads on Linux are not taken advantage of and so the Java virtual machine is not scalable when additional CPUs are added.

  On Windows JVM uses native threading.

- **Question:** When should I use notify() and when notifyAll() methods on threads?

  Answer: Generally it is more reliable to use notifyAll() method. Notify() method wakes up a single thread that is waiting on this object's monitor. If any threads are waiting on this object, one of them is chosen to be awakened. The choice is arbitrary and occurs at the discretion of the implementation. A thread waits on an object's monitor by calling one of the wait methods. Since it is internally implemented we do not know exactly which thread will be chosen.

  This method should only be called by a thread that is the owner of this object's monitor. A thread becomes the owner of the object's monitor in one of three ways:

  - By executing a synchronized instance method of that object.

  - By executing the body of a synchronized statement that synchronizes on the object.

  - For objects of type Class, by executing a synchronized static method of that class.

  Only one thread at a time can own an object's monitor.

  It is much safer to use from the beginning notifyAll() method and later trying to optimize your program you can try to switch to notify(). You have no idea which thread will be awakened. You must be sure that every waiting thread will be able to continue if you use notify() method. The behaviour of your program will be more predictable.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

From another hand using notify() method will decrease CPU and memory consumption since just one of multiple thread is awaken and rest are in waiting mode.

- **Question:** I have a piece of code (thread) which runs every x seconds and causes high CPU load. How could I decrease it?

  Answer: You can use static method yield() in the Thread. This method causes the currently executing thread object to temporarily pause and allow other threads (in OS and other programs) to execute.

- **Question:** What was wrong with stop() method for threads? Why did SUN deprecate it?
  I really need it...

  Answer: stop() method was very cruel. It stopped the thread without giving any possibility to save the data.
  It was not possible to predict when it will be applied. Even using finally method could not help - stop() interrupted it as well.
  Really it is still possible to use it. Simple solution:

  use boolean variable, for example:
  boolean letStop;
  and add two functions - prepareForStop() and doMyStop().

  Method prepareForStop() must save all your data and make sure that you run all your code in thread, then set letStop to true.
  Method doMyStop() will call stop() if only letStop value is true, otherwise do nothing (or inform you and you call prepareForStop()).
  Keep in mind that stop() is deprecated.
  --
  AP (J.A.)

- **Question:** I tried to use destroy() method on my thread. But nothing happens! I double checked my code and sure it is Ok. What is problem?

  Answer: The problem is that SUN did not implement this method. So, it exists, but does not destroy anything. Why?
  Sun says that if you use such powerful method your program will crash/hang later, maybe even sooner. I believe to it because they left stop() method even, although in deprecated form, but not destroy().

- **Question:** I know that exist Win32 and POSIX threads. Are Java threads different on different OSs? How can it affect my program?

  Answer: Although Java designed to run on different OSs it does not mean that your program will run in the same way, especially if you have threads.
  The problem is that Java runs on some OS, not Java OS. And underlying OS implementation is different on different OSs. UNIX uses POSIX threads - Portable Operating System Interface. This is standard for UNIX and approved by IEEE.
  Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Microsoft as usually made it "slightly" different that causes standard incompatibility - Win32 threads.
And implementation is different. For example, on Windows you have just limited number of priority levels, I do not remember how many, but it was something ~10-20. In UNIX - thousands!
So, if your program uses priority comparison of threads, let say priority 23453 and priority 23454 then it will be no difference on Windows. Be aware about it.
--
AP (J.S.)

- **Question:** I read this statement: "The main thread must be the last thread to finish execution. When the main thread stops, the program terminates."
Is it true?

  Answer: Absolutely wrong!
The correct one:
When you start a program, JVM creates one thread to run your program. The JVM creates one user thread for running a program. This thread is called main thread. The main method of the class is called from the main thread. If program spawns new threads from the main thread it does stop until last thread id died. Even if the main thread dies program keep running.

- **Question:** I move from C to Java now. My friend says that I still do my programming in an old way since do not use threads. But I do not see any reason to use them in my program. Could you advice where I should use them and why?

  Answer: You are right! Having threads functionality in Java does not mean that you MUST use them.
A lot of programs work better without any threads. For example, editors, calculators. Some programs run better and more reliable if they are multithreaded, for example servers, programs with repetitive, independent tasks. The final decision of course will be done by you. Multithreaded programs much more difficult to debug, avoid dead lock situation. Because threads are running often independently you can not predict exactly what happens in your program right now. Often multithreaded programs behave differently each time you start them and it is difficult to repeat problem exactly when you debug your program. So, if you have simple logic, possibly no need to run multiple tasks - try to avoid the thread usage.
--
JA

- **Question:** I know that it is possible to create threads by two different ways. What is essential difference? What is better and when?

  Answer: That's right! We can create a new thread by subclassing Thread:
public class MyBestThread extends Thread { ...

  Also we can do it by using Runnable interface. The difference is that by implementing Runnable we change just run() method keeping the possibility to subclass something more useful...
The difference between these two methods is really small and often makes no sense.
--

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

AP (J.A.)

- **Question:** Why suspend() and resume() methods were deprecated?

  Answer: These two functions were included into API with purpose to let do Garbage Collecting (GC) and debugging.
  For these two task the usage of suspend() and resume() is meaningful. But for all other things they are real disaster because suspended thread can work as a controlling thread and hold a lock. Suspending of this thread can cause hanging of program and really can be used effectively. That's why they were deprecated.
  But you can still use them if you sure that do right thing. Nobody knows how long SUN is going to keep deprecated methods and classes. Quite possible that they never will be removed.
  --
  AP (J.A.)

- **Question:** Is it true that ThreadDeath exception was "secret" and accidentally exposed to public?

  Answer: Yes it is true. The original implementation was not intended to show it to us. But due to some mistakes of SUN programmers it got out and now it is a part of API.
  Warning: "An application should catch instances of this class only if it must clean up after being terminated asynchronously. If ThreadDeath is caught by a method, it is important that it be rethrown so that the thread actually dies." from API reference for java.lang.ThreadDeath.
  --
  AP (J.A.)

- **Question:** I know how to start thread - to run run() method... But seems there is no exit() method. How do I stop my thread? Will it run forever?

  Answer: Nothing can be run forever :-). In Java there is no exit() method. It was made intentionally. The idea was that thread will exit when run method returns. So, when you reach return in your run method - your thread will exit and die.
  If you need to keep your thread working you do not let it reach the return :-)
  --
  AP (J.A.)

- **Question:** When should I use a daemon thread? Why would I use one instead of a regular thread? What is the purpose of daemon threads?

  Answer: Any Java thread can be a daemon thread. Daemon threads are service providers for other threads or objects running in the same process as the daemon thread. For example, the HotJava browser has a daemon thread, named Background Image Reader, that reads images from the file system or the network for any object or thread that needs an image.
  Daemon threads are typically independent threads within an application that provide services for other objects within that same application. The run() method for a daemon thread is typically an infinite loop that waits for a service request.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

When the only remaining threads in a process are daemon threads, the interpreter exits. This makes sense because when there are only daemon threads remaining, there is no other thread for which a daemon thread can provide a service.

To specify that a thread is a daemon thread call the setDaemon() method with a boolean parameter that is true. To determine if a thread is a daemon thread use the accessor method isDaemon().

- **Question:** Hi, Is there somebody who can tell me why my thread sleeps longer then I told him to do...

  I have a thread that has to sleep for 60000 milliseconds. But every 4, 5 minutes it sleeps for 61000 milliseconds? I have to build an application that get the time every minute, but with this sleep I can't trust the java threads.

  So can somebody tell me what is going wrong???

  Answer: Really JDK never give you warranty that will wake your thread after XXX ms.
  You can be sure only
  that your thread will not be waked up before!

  For good timing you should take another, better for real time performance, VM.
  For example PERC from Nemonics.com or something else...

- **Question:** I have created a program with a main method that instantiates and starts three threads, the first two of which are daemons. Why daemons do die when normal thread die?

- **Answer:** Because of nature of daemon threads. They are alive if exists at least one "normal user's" thread. Otherwise they die immediately.

- **Question:** Does anyone know if there is a way for two threads to find each other if they are started in two different JVM?

  In other words, I start a thread in one JVM, then I want to PipeWrite to another Thread that is doing a PipeReader. Any help would be appreciated!

  Answer: Use of Piped streams is only supported inside the same JVM. If you want IPC, you have to use sockets or xfer data using files. I would recommend the sockets approach, it has the added advantage of working not only between process, but across the network as well.

  Use of other IPC mechanisms, like shared memory, pipes, mail boxes, etc, is not supported by the JVM core. You could always roll your own native code, but why bother when you have access to sockets?

  RMI is another possibility but it all depends on what you want to do. I'll let the RMI experts talk about this option.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

And CORBA, if you want to get real fancy.
--
Joe
first published on comp.lang.java.* newsgroup

- **Question:** I would like to ask a question about garbage collection of Thread Object.

When will a Thread Object be garbage collected?
When the reference count to it becomes zero, or when it enters the "Dead" state, i.e. the run() member function terminates?

Answer, part1: Since Thread is also an Object, it will only garbage collected when the reference count is zero. You may think it is quite non-sense. the thread is useless when it enter "dead" state. why not garbage collect it?
That's because the thread object itself may contain some other useful information even the thread dead , e.g. the result of the execution of the thread. Thus, it is not sensible to do garbage collect when the reference count is not zero.
--
Anthon
P.S. Important ad! Except when object A holds a reference only to object B and object B holds a reference only to object A. Both reference counts are non-zero, but both objects are eligible for garbage collection. Which is why few, if any, modern VMs use reference counting to determine eligibility for garbage collection?
Jim

Answer, Part 2: You can consider a Thread object as a normal Object for garbage collection purposes if you consider this one rule: A running thread will not be garbage collected.
That is, a normal running thread is a root object, so the Garbage Collector will not attempt to mark it for collection. When the thread is not running, though, normal Garbage Collection rules apply (i.e. total # references == 0 then collect).

To get a thread with different behaviour,
see the Thread.setDaemon(boolean bState) method.
--Brian

- **Question:** I was until recently using the MS version of the Java API and I was using Thread.stop() when I switched to SUN I see it is deprecated.

I understand why, in fact it was causing a problem that has been solved by taking it out.
However, I do need to halt the thread from running without making it sleep.
How can I do this?

Answer: One simple way to kill a thread is to have a boolean stop variable with a method to set it to true, like so:

```
endThread(){
  stop = true;
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
run(){
    while(!stop){
        //all run code goes here
    }
}
```

If the rest of your code is okay with the last loop finishing before run() ends and the Thread dies, this works great as it is simple and can't cause any bad states. If not you can add more conditional statements just before you would affect something your stopped thread shouldn't.
After a certain point adding conditionals would become too inefficient and I'm sure there's a solution to that but I don't want to figure it out right now.
--
Ben

Ł **Question:** How do I overcome failures in threads?

How can I monitor multiple threads? I need to find what my threads do.
I have found thread.isalive() but I have a pool of threads and they are always alive. What I need is more specific info.

I have also found thread.getclass() but I didn't understand how to use it.

For example, if I write:

```
public class test implements Runnable .....{
   private int data;
   public int getData() { return data; }
}
```

Is it possible to use:

```
test test1 = (test)Thread.currentThread().getClass(); ?
```

I have multiple threads and I need to create a model which can handle the problems with threads. For example if one thread fails I need to interrupt that thread and complete its work.

Answer: Try to use such kind of inspecting pattern:

A. you have an inspector class Inspector which extends Thread and has a method:

```
public void
handleProblem(ObjectContainingAllInformationsAboutTheProblem o) {
   // here you get the information about the problem
   // from the object o and use them to handle this
   // problem.
}
```

B. You have a Pool of thread objects MyThread each of them implementing two methods :

```
public void inspect(Inspector i) {
   if (thereIsAProblem) {
      i.handleProblem(new
      ObjectContainingAllInformationsAboutTheProblem(...))
   }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

C. Your main program creates an instance of Inspector and informs it about the thread pool (gives it a reference to the pool) and starts Inspectors thread.

D. In its run() method, Inspector will periodically inspect all MyThread objects calling a method such as the following :

```
private void inspectAll() {
   // threadPool is the reference to the TreadPool...
   for (int i=0 ; i < threadPool.numberOfThreads ; i++) {
      threadPool.getThread(i).inspect(this);
   }
}
```

Well, I hope it helps you :-)

Anyway you can extends the inspectors interface to gather more information and to be able to monitor your threads in the way you want and when you want (if you don't inspect the Threads you don't get any information)
--
Michel

Ł **Question:** Anyone know how I can stop a thread without using the deprecated stop() method?

Normally I would just have a "while(alive) {...}" loop, and some "stop() {alive = false;}" method.
However, this time all the thread does is call one method, which it sits in until it finishes, and I want to stop it in certain situations. The method isn't one I've written, so I can't alter that.

Answer: I always construct my threads so they can be killed at any time. An outline of the code (this is just a sketch) is the following.

```
Mumble mumble = new Mumble(args);
Thread t = new Thread(mumble);
t.setDaemon(true);
t.start();


private class Mumble implements Runnable {
   private Thingy args = null;

   public Mumble(Thingy args) {
      this.args = args;
   }

   public void run() {
      try {
         process();
      }
      catch (InterruptedException e) {
         // cleanup processing and fall through to method end
      }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
   }

   private void process() {
      while(true) {
         // do something with args
      }
   }
}
```

Note that calling interrupt() on a Thread doesn't just instantly generate an InterruptedException.

The interrupt does happen instantly because the thread calling it is executing, not the thread executing the process() (unless you are on a multi-processor machine in which case this may not be true). The point of having interrupt() is to be able to stop something like a long running method to complete.

The only very difficult problem lies in blocking I/O operations, which do not throw InterruptedException and probably should. Blocking I/O will block until either the underlying stream is closed or the operation is complete... so be careful about using large blocking I/O operations for cancellable I/O.
--
Allan Wax

- **Question:** I want to have a program that does something every 5 seconds. Is there a method called pause(5000) or wait(5000)?
- The point is that I don't want to create a thread simply for this purpose. Thread.sleep() can have the current thread sleep/idle for certain period of time, but can the job be done without threading?

Answer: You can use Thread.sleep(5000) without creating a new Thread object because the method is static. You can look at the example below:

```
public class TestSleep {
   public static void main ( String args[] ) {
      System.out.println( "Starting" );
      for ( int i = 1; i < 101; i++ ) {
         System.out.print( i + "\t" );
         if ( i % 10 == 0 ) {
            System.out.print( "\n" );
            try {
               Thread.sleep(1000);
            }
            catch( java.lang.InterruptedException ie ) {
               ie.printStackTrace();
            }
         }
```

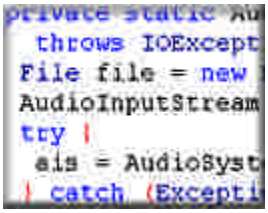
Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
      }
    }
}
```

--
Mikkel Bundgaard

22. Code Examples



Code example: I want to show you one funny thing!
The code is shown below is simplest that you can imagine and does very unusual thing! It is slightly bigger than "Hello World!" program but does much more.
It lists all files in the current directory if you run it like this:

java test *
(of course after compilation) in DOS/CMD prompt on Windows or in any shell in UNIX.
The program shows all files both in Unix and Windows.

If you do: java test .*
on UNIX it also shows all hidden files.

```
class test{
  public static void main(String args[]){
    for (int i = 0;i < args.length; i++) {
      System.out.println("File " + i + ":" + args[i]);
    }
    if (args.length<=0) {
      System.out.println("No files!");
    }
  }
}
```

You can ask how can we get this list without any file handling functionality in the code?
Indeed looks mysterious...

But in reality everything is very simple.
When you type "*" (wildcard) OS (DOS, Windows, UNIX), not Java (!!!) sends the list of files in the current directory to your program as a list of parameters. And the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

program just reads input and echoes to the screen.
And you see this list...

Question: I'm developing an application which has to get data from the server. Now, a method will go online and fetch approximately 35000 lines from the server... That can take some time, so I'd like to "show" how far the progress is. At first I had a (renamed to keep it simple) retrieveAllData() method, but it took way too long to finish, and I'd really like to constantly have a value on how far I am.

Now I've thought about having a

```
public int retrieveNextDataChunk(int count) {
  in.readLine();
  // Do some stuff.
  return count++;
}
```

that way I could have:

```
int number = 0;
while (number = retrieveNextDataChunk(number) {
  System.out.println("We're at data chunk number: " + number);
}
```

But one of my colleagues told me that he thought it was a "hack". How would you do it?

Answer: Use the Java event pattern. That is:

```
interface ProgressListener{
  public void progressMade(ProgressEvent event);
}

class ProgressEvent{
  private int chunkNumber;
  public ProgressEvent(int chunk){
     this.chunkNumber = chunk;
  }

  public int getChunkNumber(){
     return chunkNumber;
   }
}

class Transfer{
  private List progressListeners = new LinkedList();
...

  public void retrieve() {
     int count = 0;
     String nextLine;

     while ((line = in.readLine()) != null){
        // do something with line
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
            fireProgressMade(count++);
        }
      }
      public void addProgressListener(ProgressListener listener){
        progressListeners.add(listener);
      }

      public void removeProgressListener(ProgressListener listener){
        progressListeners.remove(listener);
      }

      protected void fireProgressMade(int dataChunk){
        ProgressEvent event = new ProgressEvent(dataChunk);
        Iterator listeners = progressListeners.iterator();
        while (iterators.hasNext()){
          ProgressListener next = (ProgressListener) listeners.next();
          next.progressMade(event);
        }
      }
    }
```

And the client code:

```
...
Transfer transfer = new Transfer();
transfer.addProgressListener(new ProgressListener() {
  public void progressMade(ProgressEvent event){
    System.out.println("We're at data chunk number: " +
                                                  event.getChunkNumber());
   }
});

transfer.retrieve();
// get the data and go on
```

The advantage of this over your own solution is that the public interface is not unnecessarily complicated by the presence of progress monitoring. You still just retrieve the data and use it. Now, though, if you *do* want to monitor progress, you can do it, using code that's written quite separately from the actual transfer code.

You can plug in any implementation of ProgressListener, of course, such as one that updates a progress bar in a GUI presentation. I chose one that matches the behavior you get from your short and quick solution.
--
Chris Smith

- **Question:** Can anyone answer this - basic, it seems, despite which the answer eludes me completely - **Question:** how do you have multiple windows in Java without using JDesktopPanes and JInternalFrames?

  I don't want that kind of environment. I basically want to be able to press a button/menu option to open up a small menu of options/input/buttons like the tools->internet options menu of IE, and I have no idea how to do it.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Is it actually possible without using JDPs and JIFs?

Is it as simple as creating a separate class for the menu 'mini-window' and creating an instance of it from the main system?

Answer: The example you mention is just a fancy dialog. Read documentation on Dialog/JDialog. Also a single application can instantiate and display multiple top level containers such as Frames/JFrames.

For example:

```
import java.awt.event.*;
import javax.swing.*;
public class MultiFrameTest extends JFrame{
  int x = 0;
  int y = 0;
  public MultiFrameTest(){
    super("multiFrame test");
    setSize(200,200);
    JPanel mainPanel = new JPanel();
    setContentPane(mainPanel);
    JButton addFrameBtn = new JButton("Add a frame");
    addFrameBtn.addActionListener(new ActionListener(){
      public void actionPerformed(ActionEvent e){
        addFrame(x,y);
      }
    });

    mainPanel.add(addFrameBtn);
    addWindowListener(new WindowAdapter(){
      public void windowClosing(WindowEvent we){
        System.exit(0);
       }
    });
    setVisible(true);
  }

  public void addFrame(int a, int b){
    JFrame frame = new JFrame("Frame "+a);
    frame.setSize(100,100);
    frame.setLocation( b* 10, b * 10);
    x++;
    y +=10;
    frame.setVisible(true);
  }

  public static void main(String[] args){
    new MultiFrameTest();
  }
}
--
DB
```

Ł **Question:** How do I use the DataInputStream and DataOutputStream to transfer a file from the server to the client using sockets in JAVA?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer: This will run on a single computer.

```
// Client.java
import java.net.*;
import java.io.*;
public class Client {
  public static void main(String[] args)
  throws IOException {
    InetAddress addr =
        InetAddress.getByName(null);
    System.out.println("addr = " + addr);
    Socket socket =
        new Socket("127.0.0.1", 8080);
     try {
        System.out.println("socket = " + socket);
        BufferedReader in =
           new BufferedReader(
            new InputStreamReader(
              socket.getInputStream()));
        PrintWriter out =
           new PrintWriter(
             new BufferedWriter(
                new OutputStreamWriter(
                   socket.getOutputStream())),true);
     } finally {
         System.out.println("closing...");
         socket.close();
     }
  }
}

// Server.java

import java.io.*;
import java.net.*;

public class Server{

   public static final int PORT = 8080;
   public static void main(String[] args)
    throws IOException {
       ServerSocket s = new ServerSocket(PORT);
       System.out.println("Started: " + s);
       try {
          Socket socket = s.accept();
           try {
             System.out.println(
               "Connection accepted: "+ socket);
                 BufferedReader in =
                   new BufferedReader(
                     new InputStreamReader(
                       socket.getInputStream()));
             PrintWriter out =
               new PrintWriter(
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
                new BufferedWriter(
                  new OutputStreamWriter(
                    socket.getOutputStream())),true);
              } finally {
                System.out.println("closing...");
               socket.close();
               }
            } finally {
          s.close();
        }
      }
    }
```

by Bob Randall

- **Question:** I have seen several Java pages on the net that give examples of launch the preferred application under Windows for a given URL.

  These examples, including one at JGuru, look something like this:

  Runtime.getRuntime().exec("start http://www.somecompany.com/foo.html");

  This does not work for me. I get an IOException with "error 2", whatever that is. If I try to run other apps in my path, it works fine:

  Runtime.getRuntime().exec("winhlp32.exe");

  Is there some way to get "start" to work, or are these examples full of hot air? If "start" won't work, is there another way?

  Answer: READ before these articles!!!
  To answer my own question, there is an excellent JavaWorld tip that covers this issue:

  When Runtime.exec() won't
  http://www.javaworld.com/javaworld/jw-12-2000/jw-1229-traps.html

  Basically, you have to perform an OS check and then execute the command interpreter since "start" is a command, not an executable. Under NT/2000, you would do:

  Runtime.getRuntime().exec("cmd.exe /C start
  http://www.somecompany.com/foo.html");

  The downside of this is that the command window appears briefly before the web browser opens.
  --
  Ken Carpenter
  Also! http://www.javaworld.com/javaworld/javatips/jw-javatip66.html

- **Question:** Hi, it would be appreciated if some one could tell me where I can find a Java sample code for a draggable image, i.e. using mouse left button to drag a bitmap from one location on a dialog box and drop it on another location of the same dialog box.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Answer 1:
Example:

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.Point;
import java.net.*;

import java.awt.*;
// test of dragging various components also mouse event tests

public class DragTest extends JFrame{
  int xPos;
  int yPos;
  int lastXPos;
  int lastYPos;
  boolean first = true;
  JLabel b;
  URL url;
  Image myImage;

  public DragTest(){
    super("Drag Test");
    JPanel p = (JPanel)getContentPane();
    p.setLayout(null);
    try{
       //myImage = Toolkit.getDefaultToolkit().getImage(new URL
         ("http://www.javasoft.com//images//logos//javalogo52x88.gif"));
       url = new URL
          ("http://www.javasoft.com//images//logos//javalogo52x88.gif");
    }
    catch (Exception e){System.out.println(e);}

    ImageIcon icon = new ImageIcon(url, "Here");
    System.out.println("got image "+icon.getImageLoadStatus());
    b = new JLabel(icon);
    b.addMouseListener(new MouseAdapter(){
      public void mouseClicked(MouseEvent me){
        int times = me.getClickCount();
        if ( times <= 1){
          System.out.println("Single "+me.getX()
              +" "+me.getY());
          Point here = b.getLocation();
          System.out.println("Button is at "+here.x+" "+here.y);
        }
        if (times == 2){
          System.out.println("Double");
          me.consume();
        }
        //System.out.println("Clicked = "+times);
      }
      public void mousePressed(MouseEvent me){
        System.out.println("Pressed");
        lastXPos = me.getX();
        lastYPos = me.getY();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      }
    });
    b.setEnabled(true);
    b.setSize(b.getPreferredSize());
    //b.setLocation(0,0);
    p.add(b);
    b.addMouseMotionListener(new MouseMotionAdapter(){

      public void mouseDragged(MouseEvent me){
        // b.setEnabled(false);
        Point currentPos = b.getLocation();
        int curX = currentPos.x;
        int curY = currentPos.y;

        xPos = me.getX();
        yPos = me.getY()-24;
        if (first){
           lastXPos = xPos;
           lastYPos = yPos;
           first = false;
        }
        System.out.println("y = "+yPos+"lastY
                = "+lastYPos+" "+first);
        int deltaX = xPos - lastXPos;
        int deltaY = yPos - lastYPos;
        try{
          Thread.sleep(30);
            if ((Math.abs(deltaX) < 3)&&(Math.abs(deltaY) < 3)){
              System.out.println("Made it");
              b.setLocation(curX+deltaX,curY+deltaY);
              if (Math.abs(deltaX)< 3){
                  lastXPos = xPos;
              }
            if (Math.abs(deltaY )< 3){
              lastYPos = yPos;
            }
          }
        }

        catch (Exception e){}
        // b.setEnabled(true);
        }
      });

      addMouseListener( new MouseAdapter(){
        public void mouseClicked(MouseEvent me){
          System.out.println("Frame "+me.getX()+" "+me.getY());
        }
      });

      addWindowListener(new WindowAdapter(){
        public void windowClosing( WindowEvent we){
          dispose();
          System.exit(0);
        }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      });

      setSize(200,200);
      setVisible(true);
    }

    public static void main(String[] args){
      new DragTest();


   }
  }
```

....add to PDF page only!!!

Answer 2: PDF java

http://etymon.com/pj/

IBM's approach:

http://www-106.ibm.com/developerworks/education/transforming-xml/xmltopdf/index.html

- **Question:** Do any of you know a free web host supporting JSP pages?

  Answer: check out http://www.myjavaserver.com/

  Its offering free application deployment for java(includes jsp1.1), xml, wml
  Try http://www.webappcabaret.com , jsp's , servlets and EJB1.1 hosting.

- **Question:** When settext() method is used in JButton, lots of button is wasted.
  How can I set the margin or something like that so that I can utilize all the space of the button to write something?
  Suppose I want to create a button with "Ok" as text, by default a button with some size is created with lot of space around "Ok" text being unused.
  If I reduce the button size little, the text disappears showing "..."

  How can I get around with problem? Any ideas?

  Answer: Subclass JButton and override the getInsets method. getInsets is what the layout manager calls when it wants to render the button.

  Here is a sample:

```
import java.awt.*;
import javax.swing.*;

public class TestApplet extends JApplet {
  MButton b;
  public void init() {
    getContentPane().setLayout(new FlowLayout());
    b=new MButton("OK");
    getContentPane().add(b);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }
}

class MButton extends JButton {
  public MButton(String t) {
    super(t);
  }

  public Insets getInsets() {
    return new Insets(0,0,0,0);
  }
}
```

Question: I would like to know how I can display a gif image on a normal AWT button. I need a button which displays an Image for my project.

I know that swings button can do this but I am forced to work with AWT. Can you offer any suggestions?

Answer:

```
import java.awt.*;
import java.awt.event.*;

//class to make an animated button using images.
//Written by Mark Bernard
class ImageButton extends Button implements MouseListener {
  Image i[];
  int select=1;
  int w=0;
  int h=0;
  int iw,ih;

 //The constructor requires 4 images as described below.
  // 1. Greyed out image of the button
  // 2. Normal/unselected image
  // 3. Hover image(if mouse is hovering over the button
  // 4. Pressed image
  //Please note that the image will always take up the entire
  //display of the button. If layout managers are used
  //the image will be stretched to fit the area layed out.

  public ImageButton(Image im[]) {
    super(" ");
    i=new Image[4];
    i=im;
    iw=i[0].getWidth(this);
    ih=i[0].getHeight(this);
    setSize(iw,ih);
    addMouseListener(this);
  }

  public Dimension getPreferredSize() {
    return new Dimension(iw,ih);
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public Dimension getMinimumSize() {
    return new Dimension(iw,ih);
  }

  public void setBounds(int x,int y,int width, int height) {
    w=width;
    h=height;
    super.setBounds(x,y,width,height);
  }

  public void setSize(int width,int height) {
    w=width;
    h=height;
    super.setSize(width,height);
  }

  public void setEnabled(boolean e) {
    if(e) {
      select=1;
    }
    else {
      select=0;
    }
    repaint();
    super.setEnabled(e);
  }

  public void paint(Graphics g) {
    g.drawImage(i[select],0,0,w,h,this);
  }

  public void mouseClicked(MouseEvent e) {}
  public void mouseEntered(MouseEvent e) {
     if(select!=0) {
       select=2;
       repaint();
     }
  }

  public void mouseExited(MouseEvent e) {
    if(select!=0) {
      select=1;
      repaint();
    }
  }

  public void mousePressed(MouseEvent e) {
    if(select!=0) {
      select=3;
      repaint();
    }
  }

  public void mouseReleased(MouseEvent e) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      if(select!=0) {
        select=2;
        repaint();
      }
    }
  }
```

Question: I have a method with the following signature:

public Element process(java.io.Reader reader)

I usually call this with a java.io.FileReader, to process files from local disk.
Like this:

```
Element root = null;
FileReader fr = new FileReader("config.xml");
root = process(fr);
```

Now, I need to process a file residing on a Http-server, and I have a reference to this file in a java.net.URL.

```
Element root = null;
URL configURL = new URL(http://servername/Path/config.xml);

????
root = process(??);
```

How do I convert/call my URL to a Reader object so I can call process(Reader reader)???

Answer: The following example might be helpful to you:

```
import java.net.*;
import java.io.*;

class Dag {
  static URL url = null;
  static int[] letterCount = new int[256];

  public static void main(String[] args) {
    try {
      url = new URL("http://www.orion.no");
    } catch (MalformedURLException e) { }
    try {
      letterCount = process(new
      InputStreamReader(url.openConnection().getInputStream()));
    } catch (IOException e) { }

    for (int i=0; i<256; i++) {
      if (letterCount[i]>0) {
        System.out.print((char)i+" "+letterCount[i]+",\t");
      }
    }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }

  public static int[] process(Reader reader) {
    int c = 0;
    int[] counters = new int[256];

    while (true) {
      try {
        c = reader.read(); System.out.print( (char) c );
      } catch (IOException e) { }
      if (c<0) { break;}
      counters[c] ++;
    } return counters;
  }
}
```

The program reads from the webpage at http://www.orion.no and outputs the frequencies of all (present) letters.
--
/ Lars-Ake

Question: I would like to create a internal timer with 10ms resolution. I have created the following class from java.util.TimerTask 'TimerTask' class:

```
import java.util.TimerTask;
class MyTimerTask extends TimerTask {
  private long tvalue=0;
  private long delay;
  private long old_current_time;

  public MyTimerTask (long d) {
    delay=d;
    old_current_time=System.currentTimeMillis();;
  }

  // Run timer
  public void run() {
    long current_time = System.currentTimeMillis();
    tvalue += delay;

    // Every 1s, print local timer value and relative system time value
    if (tvalue % 1000 == 0) {
      System.out.println (tvalue+" "+(current_time-old_current_time));
      old_current_time = current_time;
    }
  }

  public long get_tvalue () {
    return tvalue;
  }
}
```

I use the following main program to test my class:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
import java.util.Timer;
import java.util.TimerTask;

class MyTimerTest {
  public static void main(String args[]) {
    long d;

    // Get period, default is 1000 ms
    if (args.length != 0)
      d = Long.parseLong (args[0]);
    else
      d = 1000;

    // Init timer
    MyTimerTask rt = new MyTimerTask (d);
    Timer t = new Timer ();

    // Start it
    t.schedule (rt, 0, d);
  }
}
```

If the period is not too short, >= 100ms, the result is not so bad:

```
[pierre@ca-ol-bordeaux-6-161 test]$ java MyTimerTest 100
1000 1454
2000 1100
3000 1100
4000 1101
5000 1099
...
```

But if I use a short 10ms period it looks like timer works very slowly:

difference between n and n+1 print is 2000ms instead of 1000ms in system time:

```
[pierre@ca-ol-bordeaux-6-161 test]$ java MyTimerTest 10
1000 1994
2000 2000
3000 2000
4000 2000
5000 2000
6000 2000
...
```

Could you give me simplest example how to do print in Java? I will work out it myself :-)

Answer: Please compile and run it! It will draw empty rectangle (you see I save your inks!)

```
import java.awt.*;
public class print {
  public static void main(String args[]){
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    Frame frm = new Frame("JavaFAQ_test");
    frm.pack();
    PrintJob printJob = frm.getToolkit().getPrintJob(frm, "print", null);

    if (printJob != null) {
       Graphics grphcs = printJob.getGraphics();
       grphcs.drawRect(50, 50, 150, 100);
       grphcs.dispose();
       printJob.end();
    }
    System.exit(0);
  }
}
```

--
AP. (J.A.)

- **Advice:** I have small advice how to avoid the usage Date for measurement the time difference between two events.
  The main idea is that the Garbage Collector collecting only objects that were created by using new().

  So if you need to measure the time difference between two events use this:
  long eventOne = System.currentTimeMills();
  long diff = eventOne - System.currentTimeMills();
  Give the rest to you Garbage Collector!
  --
  Andrey S.

  P.S. This advice was sent directly to us, to info@javafaq.nu
  Have you a such? Please send!

23. More Java Tips by Dr. Heinz M. Kabutz

..... [Issue 001] - Deadlocks

Author: Dr. Heinz M. Kabutz

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to "The Java(tm) Specialists' Newsletter", a low-volume newsletter exploring experiences of using Java "in the field" (not only java.lang.reflect.Field, but that also). The reason I have sent this email to you is because you were either on a Java course presented by me, or you are a colleague, or I thought the topic might simply interest you.

Just to clear things up, as I always tell my students, my company's name is Maximum Solutions and our logo is "The Java(tm) Specialists", hence the title for the newsletter. This does not mean we are good at Java, it simply means that this is all we do (at the moment). Perhaps in future I will change my logo to "The XML Specialists", or, heaven forbid "The VB Specialists", but then I would have to be really hard-up ;-) The logo is therefore not a display of my arrogance (those of you who know me better would assure you that arrogance is definitely not one of my characteristics, neither is a bad haircut nor rolled up jeans) but is supposed to give me focus in the IT industry.

A warning I feel I have to give you is that the tricks I have discovered in Java are not always "kosher" and are used solely at your own risk. Whenever I mention something that is non-kosher I will also attach a warning label so the more experienced of you can use it and the rest not. Always make sure that your unit tests run before and after a change to a new JDK version, the days of JDK 1.1.x were extremely interesting because each version screwed up the South Africa Locale in a different way. For the past 3.5 years I've been programming exclusively in Java and I can say that it is a fascinating language. Once you dig below the surface of Java it becomes increasingly more fun and interesting to work with, and you end up wanting to exploit the limits of the JDK (last week I made a piece of code I had written with someone else 2 years ago 1000x faster). The JDK 1.3.0 source code contained in the src.jar file when you install the JDK contains a lot of Java code, 574253 lines of Java code to be exact, which is slightly less than the Java program written in Stellenbosch in South Africa where I gained most of my Java programming experience.

Deadlocks in Java

While travelling overseas I had the priviledge of spending some time helping a Java program in dire straits. It was a graphical interface that was supposed to emulate a C++ program running on a server, bla bla bla. Anyway, this application would occasionaly have a thread deadlock with the result that the screen would stop refreshing. Imagine the despair when facing an unfamiliar source base of several 10'000 lines of Java code with a deadlock SOMEWHERE in there?!? The code was well written, but even though, the task was scary. Luckily I had heard somewhere about the secret CTRL+BREAK trick that SUN smuggled into the JDK without telling anybody. If you press CTRL+BREAK while your Java program is running in a command prompt you get a stack trace of exactly what each thread is doing and by looking at the code you easily find which thread is waiting for which lock!

When you get a deadlock in your program you want to be able to reproduce it reliably as quickly as possible, so we got half the team to just concentrate on finding deadlocks that they could reproduce. In the meantime, the other half would look at

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

the stack traces and figure out why the code was deadlocking. In the process of looking we discovered something which I had heard about second-hand but had not encountered myself. Last week I heard of the same problem occuring at another company which took a lot of effort to clear up, so if your program has any GUI in it, especially Swing, it might pay to listen to this:

In Swing, all GUI components have to be changed from within the Swing thread.

This means that you cannot execute jLabel1.setText(**"blabla"**) from within any thread besides the Swing thread.

If you have change any GUI from another thread you should rather say:

```
SwingUtilities.invokeLater(new Runnable() {
  public void run() {
    jLabel1.setText("blabla");
  }
}
```

This will do the change that you requested as soon as it can, which is usually within a few microseconds. There is another call invokeAndWait which I have hardly ever used except in race-conditions.

I was under the impression that failure to use invokeLater would cause some refresh problems of a kind where some parts of the screen get refreshed and others not, I did not realise it could cause a thread deadlock (or maybe that is what everyone was talking about - refresh problem - hmpf - more like a system meltdown). Luckily you don't always have to call invokeLater whenever you change a GUI component because in some cases you already are in the Swing GUI thread, for example if a button is pressed then the ActionListener.actionPerformed method will be called from the Swing thread. However, if you provide a callback class to another class that is not part of the AWT/Swing group you will never know what thread it is coming from, so it is safest to invokeLater.

Remember that any work you do in invokeLater is going to hold up the Swing thread from repainting the window, so please don't do big database queries inside invokeLater but rather only call invokeLater for the parts of the code that are genuinely graphics related. It might pay off to bunch all the GUI related lines in your method by refactoring your code a bit. For information on refactoring look at the book with that title by Martin Fowler.

Warning Advanced:
A small optimisation is to have a class that figures out if the current thread is the Swing thread (SwingUtilities.isEventDispatchThread()) and if it is not calls SwingUtilities.invokeLater(Runnable), otherwise it calls the runnable code directly. This way you can finish everything you need to do without interference by other threads. Reason for the Non-Kosher label is that it seems you can have several event dispatch threads, I don't know under what circumstances so if you know, please

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

enlighten me. However, it almost seems like only one of them can be active at any one time.

Anyway, this is the end of the newsletter for tonight, so please let me know your comments and feedback and remember to please let me know if I'm wasting your time with this stuff then I'll not send any more to you.

With regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 002] - Anonymous Inner Classes

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the second "The Java(tm) Specialists' Newsletter", a low-volume newsletter that is aimed at in-the-trenches Java programmers and those interested in what the trenches look like nowadays. According to the latest Computing SA, the e-commerce industry in South Africa is struggling to take off because it is almost impossible to find Java programmers with any real experience. Germany seems to be experiencing a similar trend so they want to import 30'000 Indians. Aren't we working in an exciting segment of the IT market? I'm so glad I was too jung to try cash in on Y2K bugs and instead invested my time learning more of Java.

You are either on my mailing list because you begged to be added or because you did not beg to be removed, in either case, I hope you are enjoying reading this information and that it will be of benefit to you. Please let me know if there are topics that you would like discussed or if you would like to share some of your own experiences through this newsletter. This newsletter is distributed free of charge to anyone interested, so please forward it to friends and colleagues who might find the information in this newsletter useful.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

In the last newsletter I mentioned in the non-kosher section that Java makes provision for more than one GUI thread to be in the system and that I did not know if/when this feature was used. So far I have not found out what the purpose of this is or if it is still used, although I suspect it's a leftover from the Awful Windows Toolkit. Do we have a Swing expert in the house?

Anonymous Inner Classes

A feature that we did not have in the early days of Java, when all you needed to run the VM was a 80386 with 4 MB RAM and a correctly set CLASSPATH, was the concept of the inner class, specifically the anonymous inner class (heck, we didn't need that newly-fangled fancy stuff then - we were real programmers ;-). Inner classes have made it possible to code with fewer lines and in the process to obscure the code to the unenlightened but make it wonderfully elegant to those who understand. A good friend and colleague told me "One good Java programmer is better than ten bad Java programmers" which I wholeheartedly agreed with. He then went on to say "And five bad Java programmers are better than ten bad Java programmers". Inner classes have widened this divide. The typical way of using anonymous inner classes is for writing GUI event handlers, e.g.

```
button.addActionListener(new ActionListener() {
  // This is how you define an anonymous inner class
  public void actionPerformed(ActionEvent e) {
    System.out.println("The button was pressed!");
  }
});
```

The amazing thing is that we are actually defining a new class(!) while calling another method. You can virtually make new classes in all sorts of places in Java. All of this is old hat to most of you, but last week I found a new application of anonymous inner classes that I had not thought of before. But first a diversion:

A lot of books seem to suggest that if you want to create a Thread in the middle of your code you do it as shown in this code. The example is typically known as an "idiot counter", patented by Microsoft ;-), in which they make the user dream that progress is happening by changing the screen output in some way..... (I have used this technique quite successfully in the past)

```
new Thread(new Runnable() {
  public void run() {
    try {
      while (true) {
        sleep(1000); System.out.print(".");
      }
    }
    catch(InterruptedException ex) {}
  }
}).start();
```

If we look at the definition of Thread we see that it takes a Runnable as a parameter so it sort-of makes sense to create an anonymous inner class from Runnable and stick that into the parameter. However, looking more carefully inside Thread we notice that the run() method defined in Thread calls the run() method defined in Runnable, if a Runnable has been passed into the Thread constructor. Instead, it

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

would be more efficient to do the following, because we would have one less object on the heap and one less method call per Thread creation:

```
new Thread() {
  public void run() {
    try {
      while (true) {
        sleep(1000); System.out.print(".");
      }
    }
    catch(InterruptedException ex) {}
  }
}).start();
```

This way Thread itself is made into an anonymous inner class and we override the run() method so instead of Thread.run() having to check that a Runnable exists we can just execute the code in run(). It is a very small difference and I don't know why I have not seen it used before, but I suspect one of the early Java 1.1 lemmings jumped over the cliff holding a placard containing the Runnable as a parameter and all the other lemmings followed the sign ;-) To me the difference is more conceptual than actual and leads me on to the next application of inner classes:

Anyway, I now get to the real reason for another newsletter, which is an application of anonymous inner classes which I find quite useful. In Java 1.0, when we had to pass an array as a parameter to a method we did it as follows:

```
String[] temp_names = new String[3];
temp_names[0] = "Heinz";
temp_names[1] = "John";
temp_names[2] = "Anton";
universityRegistration.addNames(temp_names);
```

or, alternatively

```
String[] temp_names = { "Heinz", "John", "Anton" };
universityRegistration.addNames(temp_names);
```

In Java 1.1, SUN sneaked in a new construct so we would not need to have a temporary variable, which according to the Refactoring folklore is bad. Since JDK 1.1 we could thus say:

```
universityRegistration.addNames(
  new String[] { "Heinz", "John", "Anton" });
```

If you wanted to pass in a Collection instead of an array it would look as follows:

```
Collection temp_names = new Vector(3);
temp_names.add("Heinz");
temp_names.add("John");
temp_names.add("Anton");
universityRegistration.addNames(temp_names);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

The ability to avoid local temporary variables with arrays was always a strong deciding factor in defining interfaces to my classes because I could get away with one line of code instead of five, and the less lines of code the better. I would therefore rather define addNames(String[] names) than addNames(Collection names) even if it meant I would have to convert backwards and forwards between arrays and collections. However, with anonymous inner classes we can get the same effect seen above but with collections:

```
universityRegistration.addNames(new Vector(3)
 {{ add("Heinz"); add("John"); add("Anton"); }});
```

How does work? Very simple, say we wanted to extend Vector with our own Vector, called MyVector, that contains the three elements "Heinz", "John", "Anton":

```
public class MyVector extends Vector {
  public MyVector() {
    super(3); // to initialise it with a size of 3
    add("Heinz"); add("John"); add("Anton");
  }
}
```

The call to the super constructor always happens first, so we could re-write MyVector as follows without changing the functionality in any way:

```
public class MyVector extends Vector {
  { // initializer block
    add("Heinz"); add("John"); add("Anton");
  }
  public MyVector() {
    super(3); // to initialise it with a size of 3
  }
}
```

If we want to make an instance of an anonymous inner class we can pass the parameters directly to the super class via the parameter list of the constructor of the anonymous class. In addition, any init block denoted by {} is done AFTER the call to the super class constructor is completed, so the class MyVector could look like this:

```
Vector myVector =
  new Vector(3) { // defining anonymous inner class
  {
    add("Heinz"); add("John"); add("Anton");
  }
};
```

From here the step to addNames(new Vector(3) {{ add("Heinz"); add("John"); add("Anton"); }}); is quite simple, I've just removed the comment and bunched the curly brackets together.

The thing that amazes me most about this "discovery" is that it is such an obvious thing to do if you have to pass a collection into a method and you know the size and elements of the collection, but in 3.5 years of Java I'd never thought of it, and I bet

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

it does not occur anywhere in the 570'000 lines of JDK 1.3 code or the mega-project I worked on in South Africa.

Warning Advanced:
When you access private data members of the outer class, the JDK compiler creates package-access member functions in the outer class for the inner class to access the private members. This firstly leaves a security hole, although all data members in Java are public to the initiated ;-), and secondly used to impede performance before hotspot came along. So, if you are writing code that has to run reasonably well on non-hotspot code it would be better for performance reasons to make data members accessed by inner classes package-access.

As always, I welcome your comments and thank you for the positive feedback from last week.

With regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 003] - Logging part 1

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the third "The Java(tm) Specialists' Newsletter", a low-volume newsletter that is aimed at Java programmers who are facing the real world every day. I hope you are enjoying reading this information and that it will help you to be more productive in your work. I want to thank Java-Guru-In-The-Making Dr.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Christoph Jung for volunteering to write a piece on dynamic proxies in an upcoming newsletter. Please let me know if there are topics that you would like discussed or if you would like to share some of your own experiences through this newsletter. This newsletter is distributed free of charge to anyone interested, so please forward it to friends and colleagues who might find the information in this newsletter useful.

I drew some fire from respectable quarters that some of the code I presented in the last newsletter was a little bit on the obscure side. The Thread example was meant to illustrate the concept of making an anonymous inner class and immediately putting it to use, as opposed to just using it as a parameter. The most probable place you would find code as shown in the last newsletter would be in the unit tests of my code. If you have a method that requires as a parameter a Vector, you can use the example from the previous newsletter to succintly pass it test data. Even though the examples might seem a bit obscure at first glance, I believe that once you've seen the "Anonymous Inner Class Init Block" example, you will begin to recognise it easily enough so it will lose most of its obscurity.

If you are not writing unit test for your code, why not? There are two good books that contain information on unit tests, Refactoring by Martin Fowler and eXtreme Programming by Kent Beck. Unit tests have saved my butt a number of times, because I am not a very "good" programmer, in that I make mistakes all the time and end up messing up old working code in the process - does that sound familiar? (So, if you let me work on some of your code and you don't have unit tests, BEWARE!) When people ask me what my job is I tell them I am a professional bug writer, because that's what I'm best at. Unit tests only work if the whole development team is using the concept, much like CASE tools such as Rational Rose. I fought a losing battle 2 years ago when most of the team, on a project I was on, stopped using Rational Rose and I spent most of my time updating other developers' diagrams.

Talking of unit tests, it is possible to use the Robot (java.awt.Robot) to write unit tests for GUI's. Click on TellMeAboutTheUIRobot for more information.

Logging Part 1 of ... (haven't decided yet ;-)

While we are on the topic of things going wrong, we should always expect things to go wrong in the field and it is very important to be able to remotely figure out what went wrong and to prevent it from going wrong in future. An example of a remote fix was the Mars bug that had to be debugged and repaired remotely (not by me ;-) - the Risks digest had some interesting articles on that. With the internet being so widely accessible, we have a great challenge because our programs might fail somewhere in a remote part of this planet that might not be very pleasant for a support team to visit, for example amongst a man-eating pygmy tribe in Java (non-tm). Preferably we would ask them to email us a log of what went wrong so that we can look at the log and determine what the problem was.

When you start with a new project it is very easy to give logging too little thought, with unpleasant consequences. It is also possible to give it too much thought and design such a complex logging system that the rest of the programming team cannot wait and starts writing their own logging functions (System.out.println springs to mind). In that case you are back to the first problem of too little thought, because each developer develops his method as he goes along. This newsletter is not going to tell you which way is best, you have to decide for yourself, but it will show you some tricks and gotcha's that I've seen several times.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

In designing a logging system, there are several questions we have to face:

How do we show when the event happened?
What if the log size gets too big?
How do you switch off logging?
Are we willing to make the client code messy to improve performance?

The combination of these factors can sway your logging system in various ways, which is why I will not say which is best. I will now present the first of a series of case studies of logging systems that one might want to write:

Case Study 1:

Frank the Unix hacker has been writing a 3-tier application server that uses several third-party libraries which output logging information such as "Connected to ISAP" or "Disconnected from AISN" or "State changed to TLA". Since Frank is a hacker, he knows about JAD.EXE and is forever grateful to Pavel for making all CLASS files open-source, but he is a bit worried about rewriting the third-party code to output to his logging system because of something called "copyright law", whatever that is. The problem is that he needs to know exactly in which order things happen in respect to his own application server, so perferably the log should show all output to the console in the same order in which it appeared. Since Frank is a UNIX hacker, he remembers a little utility called "tee" that would represent a t-piece in a pipe of output so that you could send piping to two destinations.

For example:

echo hello world | tee output > output2

would send the string "hello world" to two output files, "output" and "output2".

Frank decides to write an IO stream decorator that behaves as a TeeOutputStream. (Disclaimer: I wrote my first TeePrintStream in the middle of 1997 - I have since seen the same thing published in a magazine.) If you don't understand decorators, have a look at the IO chapter of the excellent book "Thinking in Java" by Bruce Eckel or visit a good Java course.

The TeeOutputStream would look something like this:

```
// TeeOutputStream
import java.io.*;
public class TeeOutputStream extends FilterOutputStream {
  private final OutputStream out2;
  public TeeOutputStream(OutputStream out1, OutputStream out2) {
    super(out1);
    this.out2 = out2;
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public void write( int b) throws IOException {
    super.write(b);
    out2.write(b);
  }
  public void flush() throws IOException {
    super.flush();
    out2.flush();
  }
  public void close() throws IOException {
    super.close();
    out2.close();
  }
}
```

The TeeOutputStream would now be able to take two output streams and send the same information to both of them as shown in the main method.

```
// ... the main class that starts up the application server
  public static void main(String[] args) {
    PrintStream out1 = System.out;
    OutputStream out2 = new BufferedOutputStream(
        new FileOutputStream("LOG"));
    TeeOutputStream newOut = new TeeOutputStream(out1, out2);
    System.setOut(new PrintStream(newOut, true));
    out1 = System.err;
    newOut = new TeeOutputStream(out1, out2);
    System.setErr(new PrintStream(newOut, true));
    // ... some more code...
  }
```

My personal preferance would be to re-write those lines as follows, but you should do what is most comfortable to you:

```
// ... the main class that starts up the application server
  public static void main(String[] args) {
    OutputStream log = new BufferedOutputStream(
        new FileOutputStream("LOG"));
    System.setOut(
      new PrintStream(new TeeOutputStream(System.out,log),true));
    System.setErr(
      new PrintStream(new TeeOutputStream(System.err,log),true));
    // ... some more code...
  }
```

The astute among you should now be jumping up and down pointing at the glaringly obvious mistake in the above code. There is no provision for the log file growing and filling up the hard disk. Frank (correctly) assumes that his program is never going to run for a long enough period that the size of the log file would be bigger than 10 KB, but once all his bugs are removed, something should be done to prevent the big log file. The biggest log file that an advanced support man sent me was 1.1 GB in size when I used Frank's TeeOutputStream system.

Anyway, the beauty with Frank's approach is that whenever he wants to print to the log he just does a "System.out.println()" or exception.printStackTrace(). Anything

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

that goes to the console also goes to his log in the same order in which it appeared to the user. He can also switch off logging to a file by not setting the System.out PrintStream to print to the tee, or he can set his program to only print to the log.

If we look back at the questions we face with the logging system, we see that we don't have a default way of showing WHEN the event occured, although that could be added by changing the write() method of the FileOutputStream to print the date/time after each newline. The size of the log file can be monitored by implementing a different FileOutputStream. The logging to file can be switched off easily. The client code is very simple - System.out.println("Changed to state TLA");

That's the end of Frank's logging system. As you have seen, there is a lot of scope for changing it, depending on your requirements, but there is no right answer that fits all scenarios. Next week I will present another scenario with different advantages...

Please remember to forward this free newsletter to whoever you think may be interested.

With regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 004] - Logging part 2

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Welcome to the fourth issue of "The Java™ Specialists' Newsletter" and the first issue this year, or as some ignoramuses believe, this millenium. Next week Dr. Christoph Jung will tell us how the new Java dynamic proxies work, don't miss it. It's an excellent piece of writing, so if you are not subscribed to this newsletter yet, make sure you don't miss it and subscribe. Please forward this newsletter to friends, family and fauna that you think might find it useful and interesting.

[If anything, read the second-last paragraph of this newsletter, you might end up winning a free lunch :]

Logging Part 2 of ... (still cannot decide ;-)

Last year I started talking about things that go wrong in our code. I probably sent the newsletter out too close to Christmas (merry to you, btw), because I had hardly any response. Alternatively you've added me to your junk-mail-auto-delete-mail-pest filter (sniff sniff) in which case you won't read this anyway. I hope your inbox is looking emptier and that you will find time to read this newsletter. Or maybe your code always works so there's nothing to Log ;-) [naah]

Over the past years, two very knowledgeable Java programmers asked me independently how one can find out within a method where the method was invoked from. This is particularly useful when you want to print out logging information, such as where in the program the log was printed from. My answer to them was both times a typical "consulting" answer in the form of "what an interesting problem, how would you solve it?" and then I went on to ramble about using JNI, parsing the stack trace, using the Visitor pattern or writing the whole thing with Jini. i.e. I didn't have a clue!

Graciously, one of these two men, Gary Plante, sent me some code the other day which he discovered could do just that. It works by parsing a stack trace, and I took the liberty of fiddling here and there with his code, so it probably looks nothing like what you sent me, Gary :) According to my tests, this whole idea is probably a bit on the slow side. Also, to make my code easier to understand, and to avoid relying on side-effects of the PrintWriter class, I removed most of Gary's performance optimisations with the result that my code may be a little bit more robust but it is 2.5 times slower than Gary's. I have included Gary's code "as is" to show you how it could also be done. We start counting lines at different indexes, beware.

```
/**
  The stack trace starts with the following lines:
  java.lang.Throwable
     at StackTrace2.getStackTraceAtLine(StackTrace2.java:19)

  It does not make sense to get these first 2 lines so we will
  ignore them.  The number we pass into the method is the depth
  of method calls, where 0 is the current line of code, 1 is the
  line of code that called this code, 2 is the line of code that
  called the line of code that called this code, etc.

  0:    at StackTrace2Test.g(StackTrace2Test.java:10)
  1:    at StackTrace2Test.f(StackTrace2Test.java:3)
  2:    at StackTrace2Test.main(StackTrace2Test.java:17)
 */
import java.io.*;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
public class StackTrace2 {
  private static final Throwable tracer = new Throwable();
  private static final StringWriter sw =
    new StringWriter(1024);
  private static final PrintWriter out =
    new PrintWriter(sw, false);

  private StackTrace2() {} // Avoid direct creation

  public static String getCallStack(int depth) {
    synchronized (tracer) {
      if (depth < 0) throw new IllegalArgumentException();
      // skip the first 2 lines
      int lineOfInterest = depth + 3;
      // set the buffer back to zero
      sw.getBuffer().setLength(0);
      tracer.fillInStackTrace();
      tracer.printStackTrace(out);
      out.flush();
      LineNumberReader in = new LineNumberReader(
        new StringReader(sw.toString()));
      try {
        String result;
        while((result = in.readLine()) != null) {
          if (in.getLineNumber() == lineOfInterest)
            return beautify(result);
        }
      }
      catch(IOException ex) {} // we'll just return null
      return null;
    }
  }

  private static String beautify(String raw) {
    raw = raw.trim(); // we don't want any whitespace
    if (raw.startsWith("at ")) // we also cut off the "at "
      return raw.substring(3);
    return raw;
  }
}
```

You could use this information in a VERY simple Log class as follows:

```
public class Log {
   public static void it(String msg) {
      //caller of it()
      String source = StackTrace2.getCallStack(1);
      System.out.println(source + " : " + msg);
   }
}
```

An example of how this could be used is in

```
public class StackTrace2Test {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
  public void f() {
    g();
  }
  public void g() {
    Log.it("where am I now?");
    System.out.println(StackTrace2.getCallStack(0));
    System.out.println(StackTrace2.getCallStack(1));
    System.out.println(StackTrace2.getCallStack(2));
    System.out.println(StackTrace2.getCallStack(3));
    System.out.println(StackTrace2.getCallStack(-1));
  }
  public static void main(String[] args) {
    new StackTrace2Test().f();
  }
}
```

The output would look as follows:

```
StackTrace2Test.g(StackTrace2Test.java:6) : where am I now?
StackTrace2Test.g(StackTrace2Test.java:7)
StackTrace2Test.f(StackTrace2Test.java:3)
StackTrace2Test.main(StackTrace2Test.java:15)
null
Exception in thread "main" java.lang.IllegalArgumentException
        at StackTrace2.getCallStack(StackTrace2.java:22)
        at StackTrace2Test.g(StackTrace2Test.java:11)
        at StackTrace2Test.f(StackTrace2Test.java:3)
        at StackTrace2Test.main(StackTrace2Test.java:15)
```

Attached you can find the other StackTrace.java and StackTraceTest.java files which are quite a bit more difficult to understand, but at least they do not create a new LineNumberReader each time you ask it for your line number.

Attached you can find the other StackTrace.java and StackTraceTest.java files which are quite a bit more difficult to understand, but as mentioned above, they are 2.5 times faster. If performance of your logging system is a problem you can easily have a boolean in your Log class that turns this level of detailed processing off.

The best solution is probably to do this type of low-level access in JNI. I will invite the first person who sends me a correct solution that uses JNI and is really fast, to join me for a steak over lunch at the Cattle Baron (non-CapeTown residents will have to fly to Cape Town at their own expense!)

Right, I hope you found this information interesting, I always appreciate your feedback, both positive and negative.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 005] - Dynamic Proxies - Short Tutorial

Author: Dr. Christoph G. Jung

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to our fifth "The Java(tm) Specialists' Newsletter", published by Maximum Solutions, the company who may not be good at Java(tm), but Java is all we do, so we call ourselves "Java Specialists". Thank you for not unsubscribing in hordes (hmmm, I have not told you HOW to unsubscribe, so maybe that's why ;-)

Isn't it funny how employment agencies ask you ALL the things you know, like "which word processor have you used?" and "what version of Turbo Pascal do you know?" What relevance is that to today's newsletter? None.

Anyway, we've had an interesting week sorting out small Java problems and fixing bugs in my software. Many thanks to Dr. Christoph for writing this week's newsletter, which I really enjoyed reading. For this I owe you an Esplendido! Christoph started writing Java code at the end of 1999 for a very innovative ERP company in Germany called infor Business Solutions AG, and has quickly become a real fundi at writing application server code. Prior to that, he did a doctorate in Computer Science with emphasis on Robotics and Artificial Intelligence. I don't know where Java comes in, but a PhD is general education that is useful for many spheres of life, and it does not really matter what your topic was. Believe me, my PhD topic was even less useful ;-)

Anyway, enough from me, here is a "Short Tutorial to the Java2 platform Dynamic Proxy Facilities" ..... I wish we had known about this before we started pasting access control onto our 570'000 line application .....

Short Tutorial to the Java2(TM) platform Dynamic Proxy Facilities

C.G.Jung (schorsch@javaspecialists.co.za)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

A Bad Example

One the most useful design patterns when it comes to building extensible frameworks turns out to be the PROXY pattern. For example, imagine your notoriously paniqued project manager insisting on your introducing new security features into your companies existing business model:

```
/**
 * A Class representing a single salary
 */
class Salary {
  Employee employee;
  long amount;

  public long getAmount() {
    return amount;
  }
}
```

Your straightforward approach to shield the objects from exporting sensible information, such as exorbitant management salaries, to ordinary users, such as you humble worm, would be to introduce a dedicated SecurityChecker (this is not a java.lang.Security tutorial, so please excuse the following very naive code):

```
/**
 * Realises The Systems Security Policies
 */
class SecurityChecker {
  java.security.Permission manyBucksPermission;

  long check(long amount) throws SecurityException {
    if(amount>1000000)
      SecurityManager.checkPermission(manyBucksPermission);
    return amount;
  }
}
/**
 * A Modified Salary that is "Too-Much-Bucks"-Aware
 */

class Salary {
  Employee employee;
  long amount;
  // ugly reference
  static SecurityChecker sc;

  public long getAmount() {
    // isnt this the wrong place for doing that?
    return sc.check(amount);
  }
}
```

Now your project manager is happy because of being able to ask for his next salary increase. But, you (and your colleague that is the application but not the framework

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

specialist) will quite likely curse him three times when he returns with next terms project plan that includes

heavily extending the business logic (amount will no more be stored immediately, but computed by a very complex, profile-oriented function, an additional int getAmount() is needed in Salary, an Order should be implemented, etc.) expanding the successfully introduced security policies (maybe a check(Customer customer) should be added that restricts access to data associated with particular "high-sensitive" customers).

The reason for your deadly wishes is of course that in the above solution, we have heavily mixed up application data and methods (such as amount and long getAmount()) with generic access control (such as check(long amount)) Extending and updating the 200 classes with 1000 methods of application logic involves understanding of when and how to correctly apply the available security checks. Because only you as the framework pro are able to do that, you will end up with a quite suboptimal division of responsibility and code.

The Proxy Solution

In order to free your colleague from the heavy burden of the security framework and enable him to concentrate on his business, the better choice is hence to leave his Salary class alone, but rather provide him with an inherited PROXY class that intercepts any getAmount() calls to interface the security framework:

```
/**
 * proxy that behaves as a salary
 * and that performs a security check
 * before delegation
 */

class SalarySecurityProxy extends Salary {
  // this is not application, hence the reference is ok
  static SecurityChecker sc;
  // the object to which we delegate the calls to
  Salary realObject;

  public long getAmount() {
    // framework call after delegation
    return sc.check(realObject.getAmount());
  }
}
```

Now, your collegue is free to change and extend the application logic to realise project goal a (e.g., manipulating amount and getAmount()), while you can add additional independently further proxies and intercepting/ delegation calls to pursue project goal b. Strike.

If your initial application design has been smart by using mucho, mucho interfaces (you know that you should do that anyway, don˜t you?), PROXY can be even made compatible with complex inheritances:

```
/** use interfaces everywhere in your app-logic */
interface Salary {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
...
}
/** and provide proper implementations */
class SalaryImpl implements Salary{
...
}
/** as well as framework proxies */
class SalarySecurityProxy implements Salary{
...
}
/** when you extend your model ... */
interface ManagementSalary extends Salary {
...
}
/** ... afterwards ... */
class ManagementSalaryImpl extends SalaryImpl {
...
}
/** ... it's very easy */
class ManagementSalarySecurityProxy extends SalarySecurityProxy {
...
}
```

Double Strike.

The possibilities of the proxy are nearly endless and range from transparent SECURITY over LAZY RETRIEVAL, POOLING, CACHING, and TRANSACTIONAL MARKUP up to DISTRIBUTION (remember RMI stubs?). It is highly useful to combine such differently focused proxies in a CHAIN OF RESPONSIBILITY to a complex, at the same time extensible framework, e.g., SalarySecurityProxy (performs security checks) --> SalaryPersistenceProxy (retrieves the "real" salary from a database) --> Salary

Often, PROXY is used in combination with the FACTORY pattern in order to allow the transparent insertion of proxies and chains into the application logic (your colleague would˜nt be bothered by their construction at all ... no more endless discussions ... no more fruitless explanations ... must be like heaven).

Why Dynamic Proxies are needed

However, the main drawback of PROXY is that for each application class, each application method, and each additional framework functionality, you have to implement and maintain (sic!) a dedicated interceptor class/method. Depending on the output of your colleague, this can be quite a pain in the ass ... (BTW: I˜m sure that some of you already collided with the winding RMIC procedure which actually is an automated proxy-generator and -compiler not to speak of its ubiquitious stub-deployment problem)

If reminding the mostly general nature of framework code, you ask yourself whether there is not a better solution to this issue. Actually, there is. It is called DYNAMIC PROXIES and one of the wonderful invents that came to your harddisk with your JRE/JDK1.3 installation.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

DYNAMIC PROXIES heavily rely on the already matured reflection capabilities of the Java2 platform. They separate the "interception logic", that is the part of simulating an arbitrary interface, from the "invocation logic", that is the part of calling the framework before or after delegating any method call to the proper application target.

The "invocation logic", such as to connect to a method-based security checker, is realised on purpose by implementing the freshly introduced java.lang.reflect.InvocationHandler interface (more specifically, its reflection-based invoke-Method):

```
/**
 * a generic, method-based security handler
 * that is not only useful for Salaries
 */

class SecurityHandler extends InvocationHandler {
  static SecurityChecker sc;
  final Object realObject;

  /** contructor accepts the real subject */
  public SecurityHandler(Object real) {
    realObject=real;
  }

  /** a generic, reflection-based secure invocation */
  public Object invoke(Object target,
      java.lang.reflect.Method method, Object[] arguments)
      throws Throwable {
    try{
      // call framework and then reflect the app-logic
      return sc.check(method).invoke(realObject,arguments);
    } catch(java.lang.reflect.InvocationTargetException e) {
      // reconvert nested application exceptions
      throw e.getTargetException();
    }
  }
}
```

The "interception logic" is already hardwired into the 1.3 vm and the new java.lang.reflect.Proxy class. All it requires to combine it with the above SecurityHandler code is to call the static:

```
ManagementSalary salary;

ManagementSalary smellsLikeSalary=(ManagementSalary)
  java.lang.reflect.Proxy.newProxyInstance(
    Salary.class.getClassLoader(),
    new Class[] {ManagementSalary.class},
    new SecurityHandler(salary));
```

The thus constructed smellsLikeSalary object now behaves like an implementation of ManagementSalary. In fact, it *IS* an implementation of ManagementSalary whose class has been dynamically constructed at runtime, hence dynamically. You can

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

inspect it (smellsLikeSalary instanceof ManagementSalary), cast it ((Salary) smellsLikeSalary) and call methods on it (smellsLikeSalary.getAmount()). Method calls will in turn be dispatched to the invoke method of the given invocation handler (where in our example, target points to smellsLikeSalary, method points to the internal Salary.getAmount() representation, and the arguments array is empty). Return values and exceptions will be automatically converted/casted into the compatible types as given in the interface signature. That˜s it.

(No, actually that˜s not: java.lang.reflect.Proxy has a lot more advanced features and there are some tricky considerations when building proxies for multiple/conflicting interfaces - but that˜s too much for now, please refer to the API documentation for these issues.)

If PROXY has broad applicability, this holds for DYNAMIC PROXIES even more. For example, you could imagine a PersistenceHandler that hides JDBC-access to a row in a relational database table without requiring any additional object instances. For example, you could think of a JRMPRemoteHandler that implements stub-less (HHHOOOORRRAAAYYYY ... no more RMIC) RMI-access to a remote object via the TCP/IP-based Java Remote Method Protocol.

The latter usage is especially supported by DYNAMIC PROXIES, such as smellsLikeSalary actually being serializable, if the tied handler classes are. Under http://www.dreambean.com/download/rickard/SmartWorld-1.2.zip you will find executable code and documentation that extends SUN-JRMP1.2 with suitable handlers that can be distributed through naming services, files, etc. and that can be straightforwardly enriched by load-balancing, failover, clustering, and connection-pooling features.

The first (Open Source!) EJB product that uses these handlers instead of remote stubs is jBoss whose complementary server container design is indeed realised as flexible CHAIN OF RESPONSIBILITY ... Long live the (DYNAMIC) PROXIES!

Thanks for that, Christoph...

Next week will be a bit shorter and I will show you how to put method bodies in Java interfaces, which you should never need to do, IF the framework you are working within is perfect. In a less-than-perfect world such a technique can be useful, as fellow fish-chaser Niko Brummer told me. So, hold your breath until next week.

Regards

Heinz and Christoph

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 006] - Implementation code inside interfaces

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 6th issue of "The Java(tm) Specialists' Newsletter", a weekly newsletter appearing each Thursday, which explores some of the interesting parts of the Java(tm) language. My company, Maximum Solutions, has been very active with Java programming, consulting, training and mentoring. I particularly want to thank my client DataVoice (Pty) Ltd in Stellenbosch, South Africa, for giving me the opportunity to spend hours programming in Java on one of the first real Java projects in South Africa, perhaps even the biggest, with more lines of pure Java code than the JDK 1.3, and actually paying me to have fun.

Last night I worked on one of my most successful programs. It was a small 56 line program that I wrote in QuickBasic in April 1988, which my dad, who has a drinking straw manufacturing company, has used, with only minor modifications, to print thousands and thousands of labels for the boxes in which he packs his drinking straws. Perhaps one day you will see the logo "Maximum Solutions, the QuickBasic specialists!"

It's a terrible program, impossible to decipher, even contains a GOTO statement (!) but it does the job extremely reliably. I had to change his address last night (yes, it's hard coded!) and was thinking about what effort would be involved in writing the same program in Java. I would probably store all the customer details in an Access database, have autocompleting combo boxes, etc., but it would cost a fortune in time and energy to produce, far more than I could save my making it "nicer".

It is very important that as Java enthusiasts we keep an objective view of what applications lend themselves to Java and which do not. Otherwise we might find that we end up with something that is too expensive or too slow for its intended purpose. Overselling Java might benefit us in the short term, as companies scurry to hire us, but will damage us if Java gets a bad name. A Java Architect can command a salary of US$ 170'000 in USA at the moment, let's live up to the expectation and be responsible in our claims.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Now to the trick of the week, which is something you should never need to do. Please avoid doing this under all circumstances, because there are much better places to put method code than in interfaces, but I want to show you what is possible with inner classes. The call-back mechanism shown below is actually quite useful at times, especially if you want to make asynchronous database updates, but that is a topic for another newsletter.

I want to thank Niko Brummer from DataVoice for this idea, although he vehemently denies having ever really resorted to writing implementation code in an interface. Niko is a very deep person who likes to think of alternative ways of doing things, so thanks to Niko for this crazy idea :)

```
/**
  This interface contains an interfaces representing a Method and an
  interface showing the Result of the method call using the callback
  pattern.  It also contains a data member (automatically public static
  final) which is an anonymous inner class implementing our Method
  interface.
*/
public interface CodeInsideInterface {
  public interface Method {
    public void run(Result callback);
  }
  public interface Result {
    public void result(Object answer);
    public void exception(Exception problem);
  }
  Method calculateIQ =  new Method() {
    // I always write my data members as final if possible, this catches a
    // lot of bugs at compile time
    private final java.io.BufferedReader stdin =  new java.io.BufferedReader(
      new java.io.InputStreamReader(System.in));
    public void run(Result callback) {
      int iq =  100;
      try {
        System.out.print("Do you know Java (y/n)? ");
        if ("y".equals(stdin.readLine())) iq +=  20;
        System.out.print("Do you know QuickBasic (y/n)? ");
        if ("y".equals(stdin.readLine())) iq +=  20;
        System.out.print("Do you use the Basic GOTO statement (y/n)? ");
        if ("y".equals(stdin.readLine())) iq -= 30;
        System.out.print("Do you frequently use Java reflection (y/n)? ");
        if ("y".equals(stdin.readLine())) iq -= 50;
        callback.result(new Integer(iq));
      } catch(java.io.IOException ex) {
        callback.exception(ex);
      }
    }
  };
}

/**
  This test class demonstrates how to call the method on the interface.
 */
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
public class CodeInsideInterfaceTest implements CodeInsideInterface {
  public static void main(String[] args) {
    CodeInsideInterfaceTest test = new CodeInsideInterfaceTest();
    test.calculateIQ.run(new CodeInsideInterface.Result() {
      public void result(Object answer) {
        System.out.println("Your IQ is " + answer);
      }
      public void exception(Exception ex) {
        ex.printStackTrace();
      }
    });
  }
}
```

Well, there you go. Try and understand what is happening in the code, because it will teach you something about inner classes and how the callback mechanism works.

In the first newsletter I mentioned something about having multiple event queues in AWT / Swing and how I didn't know what the purpose is. Last Sunday I was pondering how I could possibly catch ALL events that occur in AWT / Swing and after playing around for a few hours managed to figure out how the event queues work. Next week I will show you the answer to my question in the first newsletter, i.e. why does AWT allow more than one event queue, how does it work, and what practical application is there... Send me email if you want to get back copies of newsletters. We are working on a web archive.

Regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 007] - java.awt.EventQueue

Author: Dr. Heinz M. Kabutz

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 8th issue of "The Java(tm) Specialists' Newsletter"! Ok, it is only the 7th issue, just testing if you're awake. This week I want to tell you the answer to my question in the first newsletter, namely, "under what circumstances is it possible to have more than one AWT event queue?" I didn't particularly go looking for an answer, but found one "by accident" while looking for a solution to another problem.

But, before I tell you how to use multiple event queues, let me bore you with a tale of why I went looking for this. A bit over a year ago, SUN released JDK 1.3 beta, and one of my colleagues, Java GUI fundi Herman Lintvelt, told me that it contained a new class called java.awt.Robot. This "Robot" could be instructed to issue native mouse events, keyboard events or take screen shots. The purpose of this Robot was to make it possible to write tests that emulated real users by issuing native actions using a platform-independent Java interface. It is difficult to find testers willing to do repetitive, boring testing, such as clicking ALL the buttons on an application each time a build is made. The Robot could be instructed to jump to a specific place on the screen and press mouse buttons and take screen shots if necessary.

Maximum Solutions quickly got stuck in and developed a testing framework around this "Robot". It is driven by scripts in which you can specify the text on the component that should be "clicked". The framework found the exact location of the component on the screen and issued a native windows click using the java.awt.Robot. Components could thus be located precisely inspite of layout managers. The code contained some magic tricks, as you might imagine, which I will not reveal (for now - maybe later, once I've sold enough copies of the framework). Luckily I had fellow Java Contractor Guild member Sydney Redelinghuys on my payroll and together we eeked information from the VM using the standard interfaces which a casual observer would not see.

The framework could be instructed to take screen shots at certain check- points of the script, or to take screen shots if it detected an error. It could find components located on different tab sheets, components not visible on the screen due to scrolling, items in combo boxes, etc.

The main "acceptance" problem we have experienced with the UIRobot is that it is quite difficult to write a comprehensive script, especially if you are not a programmer by profession. Most testers I have met struggle to write such scripts, and would prefer an automatic procedure of recording the script.

Multiple event queues

In order to make my UIRobot more acceptable to the marketplace, I needed a way in which I could unobtrusively hook myself into the AWT event system. I wanted a hotkey that users of the UIRobot could employ from any window to open the UIRobot dialog. The user could then "record" a script, "play" a script or "edit" a script, much like the macro recorder in MS Office (TM), using text on the components to locate them again. The only thing I expect the client code to do is to call

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Class.forName("**com.maxoft.ui.robot.UIRobot**");

in order to give the UIRobot class a chance to register itself.

Once that happens, I want the hotkey to be available from any frame, dialog, component, focused or not, etc. It should be a global hotkey that you can press to activate the UIRobot dialog. I had a look at the EventQueue for my first newsletter and noticed that it followed, roughly, the Chain of Responsibility design pattern. You can register a new event queue and make it responsible for handling any new events that arrive.

I saw this pattern quite quickly, but it took me 3 hours (!) to finally get it working. Had I been more careful, it should have taken not more than 5 minutes, but sometimes I am a bit slow on the uptake. One of the disadvantages of having written too much generic Java code is that I don't recognise "private" methods as an obstacle, because I can just invoke them anyway. Designers of frameworks are not necessarily very skilled at guessing how I want to extend their framework so they often make a method private instead of protected. I even went so far as to decompile SUN's implementation of java.awt.Toolkit, namely sun.awt.SunToolkit, to try and find some way of hooking into the event queue. In the end the correct and most simple way of doing this was to write a subclass of EventQueue, called MyEventQueue, and to register it as the now reigning king of the Democratic Republic of Events with the command:

```
Toolkit.getDefaultToolkit().getSystemEventQueue().push(
  new MyEventQueue());
```

The reason it took me 3 hours to figure these couple of lines out was because I overrode the postEvent() method, instead of the dispatchEvent() method, duh! An example of MyEventQueue could look like this:

```
//: MyEventQueue.java
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class MyEventQueue extends EventQueue {
  protected void dispatchEvent(AWTEvent event) {
    // the only functionality I add is that I print out all the events
    System.out.println(event);
    super.dispatchEvent(event);
  }
}
```

So, what type of functionality can you achieve with this code? You can write a global hotkey manager, you can write a recorder that generates scripts for the UIRobot or you can disable all user actions while the GUI is busy with something else. Those of you who've tried to disable GUI input have probably used the GlassPane of the JFrame which can catch all mouse events, but not keyboard shortcuts. Thanks to David Geary for that idea in his classic book on Swing!

I mentioned to one of our system engineers the possibility of using this event queue mechanism as a global hotkey manager. He got very excited and called Herman away from the company month-end barbacue to come talk to me. We had been struggling to get application-wide global hotkeys working for 3.5 years in our

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

application and one mayor customer site was holding back on a purchase because of that problem. Herman and I sat down and we came up with the GlobalHotkeyManager, where you can register any input / action combination and it matches the two for you on a global scale. Note that you need JDK 1.3 to use ActionMap and InputMap. The other parts work in JDK 1.2.

```
//: GlobalHotkeyManager.java
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class GlobalHotkeyManager extends EventQueue {
  private static final boolean DEBUG = true; // BUG? what's that? ;-))
  private static final GlobalHotkeyManager instance =
    new GlobalHotkeyManager();
  private final InputMap keyStrokes = new InputMap();
  private final ActionMap actions = new ActionMap();
  static {
    // here we register ourselves as a new link in the chain of
    // responsibility
    Toolkit.getDefaultToolkit().getSystemEventQueue().push(instance);
  }
  private GlobalHotkeyManager() {} // One is enough - singleton
  public static GlobalHotkeyManager getInstance() {
    return instance;
  }
  public InputMap getInputMap() {
    return keyStrokes;
  }
  public ActionMap getActionMap() {
    return actions;
  }
  protected void dispatchEvent(AWTEvent event) {
    if (event instanceof KeyEvent) {
      // KeyStroke.getKeyStrokeForEvent converts an ordinary KeyEvent
      // to a keystroke, as stored in the InputMap.  Keep in mind that
      // Numpad keystrokes are different to ordinary keys, i.e. if you
      // are listening to
      KeyStroke ks = KeyStroke.getKeyStrokeForEvent((KeyEvent)event);
      if (DEBUG) System.out.println("KeyStroke=" + ks);
      String actionKey = (String)keyStrokes.get(ks);
      if (actionKey != null) {
        if (DEBUG) System.out.println("ActionKey=" + actionKey);
        Action action = actions.get(actionKey);
        if (action != null && action.isEnabled()) {
          // I'm not sure about the parameters
          action.actionPerformed(
            new ActionEvent(event.getSource(), event.getID(),
              actionKey, ((KeyEvent)event).getModifiers()));
          return; // consume event
        }
      }
    }
    super.dispatchEvent(event); // let the next in chain handle event
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

Together with the GlobalHotkeyManager.java we have a test program:

```
//: GlobalHotkeyManagerTest.java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class GlobalHotkeyManagerTest extends JFrame {
  private final static String UIROBOT_KEY = "UIRobot";
  private final KeyStroke uirobotHotkey = KeyStroke.getKeyStroke(
    KeyEvent.VK_R, KeyEvent.CTRL_MASK + KeyEvent.ALT_MASK, false);
  private final Action uirobot = new AbstractAction() {
    public void actionPerformed(ActionEvent e) {
      setEnabled(false); // stop any other events from interfering
      JOptionPane.showMessageDialog(GlobalHotkeyManagerTest.this,
        "UIRobot Hotkey was pressed");
      setEnabled(true);
    }
  };
  public GlobalHotkeyManagerTest() {
    super("Global Hotkey Manager Test");
    setSize(500,400);
    getContentPane().setLayout(new FlowLayout());
    getContentPane().add(new JButton("Button 1"));
    getContentPane().add(new JTextField(20));
    getContentPane().add(new JButton("Button 2"));
    GlobalHotkeyManager hotkeyManager = GlobalHotkeyManager.getInstance();
    hotkeyManager.getInputMap().put(uirobotHotkey, UIROBOT_KEY);
    hotkeyManager.getActionMap().put(UIROBOT_KEY, uirobot);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // JDK 1.3
    setVisible(true);
  }
  public static void main(String[] args) {
    new GlobalHotkeyManagerTest();
  }
}
```

We basically match a KeyStroke (CTRL+ALT+R) with an Action. Since the action can be invoked from anywhere, we must remember to switch it off while we are handling it, otherwise it could be invoked again by mistake. Try out what happens when you don't disable the action and press the hotkey twice.

This is one of the most interesting things I've discovered in Swing and is so extremely useful that I do not understand why it is not more widely publicised. Please let me know if you've done something similar in your coding. It seems that SUN are quite good at adding new features or improving code without bothering to tell anyone, or at least not me! ;-) At more than 500'000 lines of code in the JDK 1.3, it becomes tiresome to read through it all each time a new release comes out.

By now, you have hopefully seen the value of understanding OO Design Patterns if you want to become good at Java. I have found that Java lends itself to good OO design, certainly more than C++.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

I want to thank all of you who took the time to read and respond to my newsletters, your feedback is what really makes this worthwhile.

Regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 008] - boolean comparisons

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 8th issue of "The Java(tm) Specialists' Newsletter", where we look at "in-the-field" tips and tricks used by Java professionals. My intention with this newsletter is to have some fun spreading knowledge of advanced Java concepts, and in the process learning new things, so please keep on forwarding your ideas, comments and criticisms.

A special welcome to those of you who found out about this newsletter through Bruce Eckel's website, and a very hearty "thank you" to Bruce, the author of the best Java learning book of all time!

Comparing boolean variables with true or false

Java was based on ideas taken from C/C++, but the designers of the language tightened up the language significantly. In C/C++, a zero (0) means false and non-

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

zero (e.g. 1) means true. This caused a lot of bugs that were not found at compile time, especially with assignment vs. comparison.

Here's a snippet of C++ code, which has the same syntax as Java

```
int i = someNumber;
if (i == 4) {
  /* do something */
}
```

In C/C++ it is extremely easy to write

```
int i = someNumber;
if (i = 4) { // note we are doing assignment, not comparison!
  /* do something */
}
```

This will have the effect of assigning 4 to "i", regardless of what "someNumber" was equal to, the result will be non-zero, i.e. true, and after this code, "i" would equal 4.

In Java, we now have the "boolean" type which is either "true" or "false". The parameter to the "if" statement has to be "boolean", rather than an int as in C/C++. The code assigning 4 to i would not compile in Java. This is great, because we find a lot of bugs at compile time.

Warning Advanced:
Memory optimisation insider trick: Incidentally, when you have a boolean data member, according to my experiments, it is counted as 4 bytes! A boolean array uses one byte per boolean.

This is all very fundamental, so why am I writing this in an advanced Java newsletter? When we assign one boolean to another, the returned value is that of the value we are assigning to. This means that the bug mentioned above is still possible if we are doing boolean comparisons! I have often seen code such as the following, even from experienced, otherwise good, Java programmers (I counted over 150 occurance in the project I'm currently working in):

```
boolean pentiumTMcpu = Utils.isCpuAPentium();
if (pentiumTMcpu == true) {
  /* work out incorrect salary using double */
}
```

This will compile fine in Java, but so will the following, which assigns true to pentiumTMcpu and will always work out the salary using the Pentium bug (younger readers would not remember):

```
boolean pentiumTMcpu = Utils.isCpuAPentium();
if (pentiumTMcpu = true) {
  /* this code will always be executed */
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Instead, it would be a lot safer to write

```
boolean pentiumTMcpu = Utils.isCpuAPentium();
if (pentiumTMcpu) {
  /* work out incorrect salary using double */
}
```

It is very easy for a bug to slip in during maintenance so we should always think about how we can reduce the possibility of bugs being introduced by less experienced, less expensive, less intelligent, less careful future programmers who have to maintain the code we write. (I'm kidding about the less expensive.)

There is a technique used alot in Microsoft's C++ libraries in which they would have written the comparison as:

```
boolean pentiumTMcpu = Utils.isCpuAPentium();
if (true == pentiumTMcpu) {
  /* work out incorrect salary using double */
}
```

This is also quite safe, since you cannot assign anything to true, and so you would get a compiler error if you only had one "=". I personally don't like that style, but it is just as safe as writing "if (pentiumTMcpu)", except that you might confuse someone who comes to Java having never written C/C++.

That's the end of this newsletter. I am trying to keep these newsletters short, but sometimes get carried away with my story telling, please forgive me. I generally have topics lined up one month in advance, but if there is something you would like to contribute, please send me an email.

Next week I will look at a topic I call "depth-first-polymorphism", brought to my attention by Dr. Jung, who wrote an earlier newsletter.

Until next week

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 009] - Depth-first Polymorphism

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 9th issue of "The Java(tm) Specialists' Newsletter", where we look at "in-the-veld" tips and tricks used by Java professionals. I want to thank all of you who respond to these letters, your comments make the late nights writing these newsletters worthwhile :) And those of you who in their day-to-day communication are imitating the style of these newsletters - you know who you are ! - keep trying ;-)

I must apologize that this newsletter is quite long. Unfortunately I am under severe time pressure at the moment so I did not have time to make it shorter.

I was told after last week's newsletter that Java was not based only on C++ and Smalltalk, but also on Lisp ?!? (I did not know that, no wonder it's so terribly slow. I'm surprised the JDK doesn't come with the Emacs editor.) Anyway, I have to thank Michael Wolber from Infor AG in Germany for pointing that out and for his excellent contribution:

Greenspun's Tenth Rule of Programming:*"any sufficiently complicated C or Fortran program contains an ad hoc informally-specified bug-ridden slow implementation of half of Common Lisp."*

And Java?

Depth-first Polymorphism (or Customised Polyseme)

Consider the following class:

```
public class Polyseme {
  public static class Top {
    public void f(Object o) {
      System.out.println("Top.f(Object)");
    }
    public void f(String s) {
      System.out.println("Top.f(String)");
    }
  }
  public static void main(String[] args) {
    Top top = new Top();
    top.f(new java.util.Vector());
    top.f("hello");
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    top.f((Object)"bye");
  }
}
```

Java looks for the method with the "narrowest" matching class for the parameter objects. Therefore, the output from running this class is:

```
Top.f(Object)
Top.f(String)
Top.f(Object)
```

In Java, the virtual machine tries to find a matching method for your parameters, starting at the top of the hierarchy and moving down. Say we have the following classes:

```
public class BreadthFirst {
  public static class Top {
    public void f(Object o) {
      System.out.println("Top.f(Object)");
    }
  }
  public static class Middle extends Top {
    public void f(String s) {
      System.out.println("Middle.f(String)");
    }
  }
  public static void main(String[] args) {
    Top top = new Middle();
    top.f(new java.util.Vector());
    top.f("hello");
    top.f((Object)"bye");
  }
}
```

The virtual machine will thus start at Top and check if there are any methods which would accept String.class or Object.class, and indeed, Top.f(Object) would handle all those parameters. The output is therefore the following:

```
Top.f(Object)
Top.f(Object)
Top.f(Object)
```

We could "fix" this by overriding f(Object) and using instanceof to call the correct f() method (brrr - I'd rather get stuck on the N2 than do that [for those not living in Cape Town, the N2 is notoriously dangerous, you either get shot at or in or with if your car breaks down])

```
public class BreadthFirstFix {
  public static class Top {
    public void f(Object o) {
      System.out.println("Top.f(Object)");
    }
  }
  public static class Middle extends Top {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    public void f(Object o) {
      if (o instanceof String)
        f((String)o);
      else
        super.f(o);
    }
    public void f(String s) {
      System.out.println("Middle.f(String)");
    }
  }
  public static void main(String[] args) {
    Top top = new Middle();
    top.f(new java.util.Vector());
    top.f("hello");
    top.f((Object)"bye");
  }
}
```

The output would now look as we would expect:

```
Top.f(Object)
Middle.f(String)
Middle.f(String)
```

This might have the correct effect, but it does mean that we have to have such a silly "instanceof" in all the subclasses. If we are designing a OO framework we want to have our clients subclass our classes without having to do acrobatics to achieve this.

Christoph Jung mentioned this problem with Java to me a few weeks ago and we thought of some code you could put at the highest level class that uses reflection to start at the lowest class and then tries to match the method to the type before moving up the hierarchy. I call this "depth-first-polymorphism".

```
import java.lang.reflect.*;
public class DepthFirst {
  public static class Top {
    private Method getPolymorphicMethod(Object param) {
      try {
        Class cl = getClass();  // the bottom-most class
        // we start at the bottom and work our way up
        Class[] paramTypes = {param.getClass()};
        while(!cl.equals(Top.class)) {
          try {
            // this way we find the actual method
            return cl.getDeclaredMethod("f", paramTypes);
          } catch(NoSuchMethodException ex) {}
          cl = cl.getSuperclass();
        }
        return null;
      }
      catch(RuntimeException ex) { throw ex; }
      catch(Exception ex) { return null; }
    }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    public void f(Object object) {
      Method downPolymorphic = getPolymorphicMethod(object);
      if (downPolymorphic == null) {
        System.out.println("Top.f(Object)");
      } else {
        try {
          downPolymorphic.invoke(this, new Object[] {object});
        }
        catch(RuntimeException ex) { throw ex; }
        catch(Exception ex) {
          throw new RuntimeException(ex.toString());
        }
      }
    }
  }
  public static class Middle extends Top {
    public void f(String s) {
      System.out.println("Middle.f(String)");
    }
  }
  public static class Bottom extends Middle {
    public void f(Integer i) {
      System.out.println("Bottom.f(Integer)");
    }
  }
  public static class RockBottom extends Bottom {
    public void f(String s) {
      System.out.println("RockBottom.f(String)");
    }
  }
  public static void main(String[] args) {
    Top top = new RockBottom();
    top.f(new java.util.Vector());
    top.f("hello");
    top.f(new Integer(42));
    top = new Bottom();
    top.f(new java.util.Vector());
    top.f("hello");
    top.f(new Integer(42));
  }
}
```

The answer is this time:

```
Top.f(Object)
RockBottom.f(String
Bottom.f(Integer)
Top.f(Object)
Middle.f(String)
Bottom.f(Integer)
```

When should you use this technique? Only if you have a lot of specific type handlers as subclasses of a common superclass where it would make sense to add such a depth-first invoker. You can probably extract this functionality and put it in a

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

separate class. If you use this commercially please do the exception handling correctly, I didn't bother in my example, in preparation for when I change my logo to "The C# Specialists".

Thanks for your comments, I always appreciate your feedback.

Regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 010] - Writing GUI Layout Managers

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 10th issue of "The Java(tm) Specialists' Newsletter". It has been a lot of fun writing these newsletters, especially since there is soooo much to write about in Java. Today I heard about some cases where in the JDK 1.3 the GUI threads get started which were not started in JDK 1.2.2. The effect is that command-prompt driven server code suddenly did not shut down anymore. An example is at the end of this newsletter thanks to James Pereira, Technical Director of Foundation Technologies, one of very few directors who still finds time to actually write Java code. He promised me a newsletter on class versioning with multiple ClassLoaders which should be very interesting. These differences in Java VMs harks of the days when we used to use C macros (#ifdef) to distinguish between versions of a 8087 FP chip used on a specific machine. At the rate Java gets changed, we have to be VERY careful to not rely on the existence of certain bugs.

But now, let's go back a few years, when JBuilder 2.0 had just come out. In those days, the GridBagLayout was not supported as well as in the later JBuilder versions, so we tried to avoid using it where possible. If you want to age quickly, try and maintain some GridBagLayout code written by someone else ...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

A colleague of mine wanted to do a screen layout similar to BorderLayout but with different rules. The rules for BorderLayout are that the layout manager allocates preferred height for North and South components, then preferred width for East and West components, and the rest goes to Center. He wanted to have Left, Middle and Right components where the Middle took on its preferred width and Left and Right took on the remaining width shared equally. It would look like this:

```
OOOOOOOOOO    OOOOOOOOOO
OOOOOOOOOO    OOOOOOOOOO
OOOOOOOOOOaaaaOOOOOOOOOO
OOOOOOOOOOaaaaOOOOOOOOOO
OOOOOOOOOOaaaaOOOOOOOOOO
OOOOOOOOOO    OOOOOOOOOO
OOOOOOOOOO    OOOOOOOOOO
```

if we resized the screen, it would look like this:

```
OOOOOOOOOOOOOO    OOOOOOOOOOOOOO
OOOOOOOOOOOOOO    OOOOOOOOOOOOOO
OOOOOOOOOOOOOO    OOOOOOOOOOOOOO
OOOOOOOOOOOOOOaaaaOOOOOOOOOOOOOO
OOOOOOOOOOOOOOaaaaOOOOOOOOOOOOOO
OOOOOOOOOOOOOOaaaaOOOOOOOOOOOOOO
OOOOOOOOOOOOOO    OOOOOOOOOOOOOO
OOOOOOOOOOOOOO    OOOOOOOOOOOOOO
OOOOOOOOOOOOOO    OOOOOOOOOOOOOO
```

or

```
OOOOOOO    OOOOOOO
OOOOOOOaaaaOOOOOOO
OOOOOOOaaaaOOOOOOO
OOOOOOOaaaaOOOOOOO
OOOOOOO    OOOOOOO
```

We tried the standard way of combining layout managers with panels but were unable to do this layout. In those days without decent GridBagLayout tool support we were very hesitant to hack around with that monster layout. (Incidentally, the same type of layout is used by the jGuru crew to demonstrate that you need the GridBagLayout, so they also did not seem to get it right with normal layout managers.)

This caused me to try and write my own Layout Manager, which I called the WildLayoutManager, named after my ex-colleague, who is.

Writing your own GUI Layout Manager

It is actually increadibly straightforward writing your own layout manager. This layout manager took me one lunch time, i.e. less than 1 hour. Permanent employees at that company get free lunches, but as a contractor I had to pay, so I programmed instead of ate in those days. Nowadays I simply talk about work during lunch and charge the time ;-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Seriously, it really is easy. When I tell people that I've written my own LayoutManager to do the layout for me they get all boggle-eyed and start holding up crucifixes, garlic or search for the silver bullet. (Those in the know, know that there is no silver bullet for software development.)

```
//: WildLayoutManager.java
import java.awt.*;

public class WildLayoutManager implements LayoutManager {
  // these are the constraints possible with the WildLayoutManager
  public static final String LEFT = "Left";
  public static final String RIGHT = "Right";
  public static final String MIDDLE = "Middle";

  // We keep handles to three components, left, right and middle
  private Component left;
  private Component right;
  private Component middle;

  // we need to be able to add components.  if two components are added
  // with the same constraint we keep the last one
  public void addLayoutComponent(String name, Component comp) {
    if (LEFT.equals(name)) {
      left = comp;
    } else if (RIGHT.equals(name)) {
      right = comp;
    } else if (MIDDLE.equals(name)) {
      middle = comp;
    } else {
      throw new IllegalArgumentException(
        "cannot add to layout: unknown constraint: " + name);
    }
  }

  // here we remove the component - first find it!
  public void removeLayoutComponent(Component comp) {
    if (comp == left) {
      left = null;
    } else if (comp == right) {
      right = null;
    } else if (comp == middle) {
      middle = null;
    }
  }

  // The minimum dimension we're happy with is the preferred size
  // this could be more fancy by using the minimum sizes of each component
  public Dimension minimumLayoutSize(Container parent) {
    return preferredLayoutSize(parent);
  }

  // Here we work out the preferred size of the component, which is used
  // by methods such as pack() to work out how big the window should be
  public Dimension preferredLayoutSize(Container parent) {
    Dimension dim = new Dimension(0, 0);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    // get widest preferred width for left && right
    // get highest preferred height for left && right
    // add preferred width of middle
    int widestWidth = 0;
    int highestHeight = 0;
    if ((left != null) && left.isVisible()) {
      widestWidth = Math.max(widestWidth, left.getPreferredSize().width);
      highestHeight =
        Math.max(highestHeight, left.getPreferredSize().height);
    }
    if ((right != null) && right.isVisible()) {
      widestWidth = Math.max(widestWidth, right.getPreferredSize().width);
      highestHeight =
        Math.max(highestHeight, right.getPreferredSize().height);
    }
    dim.width = widestWidth * 2;
    dim.height = highestHeight;
    if ((middle != null) && middle.isVisible()) {
      dim.width += middle.getPreferredSize().width;
      dim.height = Math.max(dim.height, middle.getPreferredSize().height);
    }

    Insets insets = parent.getInsets();
    dim.width += insets.left + insets.right;
    dim.height += insets.top + insets.bottom;

    return dim;
  }

  // this is the brain of the layout manager, albeit rather small.
  // I told you this is straightforward...
  public void layoutContainer(Container target) {
    // these variables hold the position where we can draw components
    // taking into account insets
    Insets insets = target.getInsets();
    int north = insets.top;
    int south = target.getSize().height - insets.bottom;
    int west = insets.left;
    int east = target.getSize().width - insets.right;

    // we first find the width of the left and right components
    int widestWidth = 0;
    if ((left != null) && left.isVisible()) {
      widestWidth = Math.max(widestWidth, left.getPreferredSize().width);
    }
    if ((right != null) && right.isVisible()) {
      widestWidth = Math.max(widestWidth, right.getPreferredSize().width);
    }
    if ((middle != null) && middle.isVisible()) {
      widestWidth = Math.max(widestWidth,
        (east - west - middle.getPreferredSize().width) / 2);
    }

    // next we set the size of the left component equal to the widest width
    // and whole height, and we set the bounds from North-West corner
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    if ((left != null) && left.isVisible()) {
      left.setSize(widestWidth, south - north);
      left.setBounds(west, north, widestWidth, south - north);
    }
    // next we set the size of right component equal to the widest width
    // and whole height, and we set the bounds from North-East corner
    if ((right != null) && right.isVisible()) {
      right.setSize(widestWidth, south - north);
      right.setBounds(east-widestWidth, north, widestWidth, south - north);
    }
    // lastly we set the size of the middle component equals to the
    // remaining width, which should be equal to the middle object's
    // preferred width and we set the height equal to the middle object's
    // preferred height
    if ((middle != null) && middle.isVisible()) {
      middle.setSize(east - west - widestWidth * 2,
        middle.getPreferredSize().height);
      middle.setBounds(
        west+widestWidth,
        north + (south - north - middle.getPreferredSize().height)/2,
        east - west - widestWidth * 2,
        middle.getPreferredSize().height);
    }
  }
}
```

You see, it really was quite simple. Here is an example frame that tries out the new layout manager:

```
//: WildLayoutExample.java
import java.awt.*;
import javax.swing.*;

public class WildLayoutExample extends JFrame {
  public WildLayoutExample() {
    super("WildLayoutExample");
    setSize(new Dimension(400, 300));
    getContentPane().setLayout(new WildLayoutManager());
    // construct the left panel
    JPanel leftPanel = new JPanel(new BorderLayout());
    leftPanel.add(new JLabel("Left Label"), BorderLayout.NORTH);
    leftPanel.add(new JTree(), BorderLayout.CENTER);
    // construct the middle panel
    JPanel middlePanel = new JPanel(new GridLayout(0,1,5,5));
    middlePanel.add(new JButton("Add >"), null);
    middlePanel.add(new JButton("<< Remove All"), null);
    // construct the right panel
    JPanel rightPanel = new JPanel(new BorderLayout());
    rightPanel.add(new JLabel("Right Label"), BorderLayout.NORTH);
    rightPanel.add(new JTextArea("jTextArea1"), BorderLayout.CENTER);
    // add the panels to the content pane using our new layout manager
    getContentPane().add(leftPanel, WildLayoutManager.LEFT);
    getContentPane().add(middlePanel, WildLayoutManager.MIDDLE);
    getContentPane().add(rightPanel, WildLayoutManager.RIGHT);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }
  public static void main(String[] args) {
    WildLayoutExample frame = new WildLayoutExample();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // JDK 1.3 !
    frame.setVisible(true);
  }
}
```

You could use the idea of custom layout managers to create all sorts of interesting layouts, such as a special form layout for designing forms for your business application. I do not see any particular problems with writing your own layout manager, especially if there is some layout that you want to use quite often. Just don't use absolute layouts, whatever you do!!!

And remember: let's be careful out there!

Heinz

Warning Advanced:
The following code exits normally under JDK 1.2.2 but "hangs" under JDK 1.3. Many thanks to James Pereira for pointing this out and sending me the code:

```
import java.awt.Cursor;
public class J23D {
  // This cursor class when loaded now starts the GUI event threads!
  private static Cursor s_waitCursor =
    Cursor.getPredefinedCursor(Cursor.WAIT_CURSOR);
  public void log(String _msg) {
    System.out.println(_msg);
  }
  public static void main ( String[] _args ) {
    new J23D().log("That's all folks!");
  }
}
```

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 011] - Hooking into the shutdown call

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 11th issue of "The Java(tm) Specialists' Newsletter", where we have a "gloves-off" approach to the Java(tm) language and development environment. Thank you again for your support and the feedback you send me.

I am sitting outside in a mild South African summer evening, looking up at the Southern Cross hoping the milky way will inspire me to bring you a newsletter that is worth not pressing delete on. I am always interested in hearing about "gotchas" that you found while using Java, so please keep on sending me your ideas.

Since the GUI is 63% of the JDK, I was tempted to send another interesting bit about setting the focus to a component besides the first one on a modal dialog. This seemingly simple problem had me quite stumped, but luckily the Cape Town Java User Group gave me one solution and I found another solution myself. Next week I will write about that, so stay tuned. I know that "real" programmers don't write GUIs, but if you are a "real" Java programmer, other less real programmers might ask you how to do these fancy tricks, and it's good to then have an answer ;-) There is another newsletter in the pipeline that tells you how to serialize GUI components across the network.

Hooking into the shutdown call

Last year, while I was trying out the jBoss EJB application server, I happened to notice that you could shutdown the server by pressing CTRL+C. Somehow, the server "knew" you had killed the program, and shutdown all the necessary services before exiting the Java VM.

My Java-Guru-In-The-Making colleague told me that it was possible to add a "shutdown hook" which would be called when the application was being shutdown. This shutdown event would happen when you call System.exit(), which in turn gets called when you press CTRL+C. As was hinted in last week's newsletter, once you use any GUI components, the VM starts up GUI threads that you cannot shutdown cleanly. This means that the only option you have of exiting your program is to call System.exit()!!!

However, System.exit() can be called from anywhere in your program and if it gets called, we want to have a central place where we can add methods that will be called when we want to exit. This central place was added, without much fanfare (actually

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

no fanfare at all!), in JDK 1.3, together with many other cool features like the java.awt.Robot.

Here is an example of how you could add a shutdown hook into the current VM runtime. You do this by constructing a new Thread and passing that to the runtime to start when it wants to shutdown.

```
public class ShoutdownTrick {
  public static void main(String args[]) {
    Runtime.getRuntime().addShutdownHook(new Thread() {
      public void run() {
        System.out.println("You wanna quit, hey?");
        System.out.println("... fry eggs on your CPU.");
        while(true);
      }
    };
    System.out.println("Let's take a break...");
    try {
      Thread.sleep(5000);
    }
    catch(InterruptedException ex) {}
    System.out.println("That's it, I'm outta here");
    System.exit(0);
    System.out.println("This line will not show!");
  }
}
```

Try run this on your machine and you will notice that the only way to stop the program is via the task manager. This of course highlights one of the problems with writing code to call during shutdown. What happens when the shutdown code does not exit? I am told that in Delphi you can solve the Halting Problem, unfortunately Java is not that powerful, so the VM cannot know whether the shutdown code will stop or not. Ideally, you should in the shutdown code only do what is absolutely necessary to put the system in a state that is usable next time you want to run the program, i.e. close DB connections cleanly, close log files, etc.

A feature I did not know about until tonight is that you can bypass the shutdown hooks and actually "halt" the VM without further ado. So, the obvious next step in our Shutdown Hook example is to add another thread that will halt the VM after some timeout:

```
public class ShoutdownTrickWithHaltTimeout {
  public static void main(String args[]) {
    Runtime.getRuntime().addShutdownHook(new Thread() {
      public void run() {
        System.out.println("You wanna quit, hey?");
        System.out.println("... fry eggs on your CPU.");
        while(true);
      }
    });
    Runtime.getRuntime().addShutdownHook(new Thread() {
      public void run() {
        try {
          Thread.sleep(10000);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      } catch(InterruptedException ex) {}
      // halt will bail out without calling further shutdown hooks or
      // finalizers
      Runtime.getRuntime().halt(1);
    }
  });
  System.out.println("Let's take a break...");
  try {
    Thread.sleep(5000);
  }
  catch(InterruptedException ex) {}
  System.out.println("That's it, I'm outta here");
  System.exit(0);
  System.out.println("This line will not show!");
 }
}
```

The problem with this code is that even though you know that after 10 seconds you will definitely halt the VM, you will have to wait for 10 seconds each time you try to exit the VM, because there is no way of knowing if the other threads are finished yet. This brings out another interesting feature, which is that all the shutdown hooks are threads and it seems that they are all started at the same time. The obvious solution is to make the "Halting" shutdown thread a Daemon thread, but, alas, at that point in the VM processing, it doesn't matter whether a thread is a Daemon thread or not, it will run until it is finished.

It seems that, even though our friends at Sun have noticed that such a shutdown hook feature is necessary, they have not fully thought through the possibilities of what could go wrong and how to solve it. Perhaps a System.exit(int status, long timeout) would be appropriate?

There are some really cool extras in JDK 1.3 which are not very widely publicised, or perhaps I'm reading the wrong magazines?

Shall we take a bet how long it takes before all the System.exit(0) code in the world is replaced with Runtime.getRuntime().halt(0)?

Until next week

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 012] - Setting focus to second component of modal dialog

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 12th issue of "The Java(tm) Specialists' Newsletter". Please forward this free newsletter to as many people as you know who might be interested in "advanced" Java topics. You are welcome to send me questions on topics in my newsletters, I will do my best to answer them.

The code in these newsletters has been tested using JDK 1.3.

Setting focus to second component of modal dialog

A few weeks ago I got stumped by a seemingly simple problem. I was trying to write a login dialog that would remember the last username entered and put the focus on the password field if an old username was found. I battled against the tide of Swing, even posted a question to the local Java User Group mailing list, but eventually I performed some obscure tricks to conquer this basic beginner's problem.

Warning Advanced:
A problem with dialogs is that they are very often not bound to a parent frame, especially modal dialogs. This is not very good, because if you move to another application and move back to your Java application via the task bar in Windows, you cannot see the dialog. This single "bug" has caused a lot of confusion for users who think their Java application has hung up, but if they ALT+TAB to the application they can see the dialog again. A good solution is to create a frame at position -1000, -1000 and use that as the owner if the dialog does not have an owner. It is also possible to write a class which works out when a new window is shown and maps the title to the frame. This way you can find existing frames given a title. No, I won't tell you in this newsletter how to do that, no space.

My LoginDialog looked something like this:

```
//: LoginDialog.java
import javax.swing.*;
import java.awt.*;
public class LoginDialog extends JDialog {
  private final JTextField userName = new JTextField(8);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  private final JPasswordField password = new JPasswordField(8);
  public LoginDialog(Frame owner) {
    super(owner, "Login Dialog", true);
    getContentPane().setLayout(new GridLayout(0,2,5,5));
    getContentPane().add(new JLabel("Username:"));
    getContentPane().add(userName);
    getContentPane().add(new JLabel("Password:"));
    getContentPane().add(password);
    pack();
    Windows.centerOnScreen(this);
    show();
  }
  public String getUserName() { return userName.getText(); }
  public String getPassword() { return password.getText(); }
  public static void main(String[] args) {
    JFrame owner = new JFrame("Login Dialog");
    owner.setLocation(-1000, -1000);
    owner.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    owner.show();
    new LoginDialog(owner);
  }
}
//: Windows.java
import java.awt.*;
public class Windows {
  public static void centerOnScreen(Window window) {
    Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
    window.setLocation(
      (d.width - window.getSize().width) / 2,
      (d.height - window.getSize().height) / 2);
  }
}
```

As mentioned before, I wanted my focus to start on the password field, rather than the user name field. So, the obvious place to set the focus is after the call to "centerOnScreen", i.e. change the code to

```
// ...
  pack();
  centerOnScreen(this);
  password.requestFocus();
  show();
}
// ...
```

Unfortunately, you can only change the focus to components which are visible on the screen, and since the dialog has not been shown yet, trying to set the focus has no effect.

The obvious solution to this problem is to request the focus after the show() has been called. But, since this is a modal dialog, show will only return once the dialog has been closed, so even though the component is now visible, we will only request focus once we have closed the dialog, which does not help us awefully much.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Again, the seemingly obvious solution to this problem is to call the requestFocus method using SwingUtilities.invokeLater(), but you are not guaranteed that the dialog will then be visible, and if it is not, you again have no effect. You could of course wait for 10 seconds and then request focus, but that would result in a rather awkward user interface.

I posted this problem to a local Java user group and got one response to how this could be solved. But first I will show you my solution, which is terribly obscure, but I could not come up with anything better. Please send me your solutions if they differ from these:

Solutions 1

We want to pass the focus on as soon as we get the focus in the username field. We thus add a focus listener to the userName field, which transfers the focus to the next component when the focusGained method is called. We only want to do that when the dialog is constructed, so when the focusLost method is called we remove the listener again. LoginDialog would now look like this:

```
//: LoginDialog2.java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class LoginDialog2 extends JDialog {
  private final JTextField userName = new JTextField(8);
  private final JPasswordField password = new JPasswordField(8);
  public LoginDialog2(Frame owner) {
    super(owner, "Login Dialog", true);
    getContentPane().setLayout(new GridLayout(0,2,5,5));
    getContentPane().add(new JLabel("Username:"));
    getContentPane().add(userName);
    getContentPane().add(new JLabel("Password:"));
    getContentPane().add(password);
    pack();
    Windows.centerOnScreen(this);
    userName.addFocusListener(new FocusListener() {
      public void focusGained(FocusEvent e) {
        userName.transferFocus();
      }
      public void focusLost(FocusEvent e) {
        userName.removeFocusListener(this); // refers to listener
      }
    });
    show();
  }
  public String getUserName() { return userName.getText(); }
  public String getPassword() { return password.getText(); }
  public static void main(String[] args) {
    JFrame owner = new JFrame("Login Dialog");
    owner.setLocation(-1000, -1000);
    owner.show();
    owner.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    new LoginDialog2(owner);
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

Yes, it is fairly obscure, but so is solution # 2, given to me by my "Bruce Eckel Handson" student, Charl Smit from CCH in South Africa. Thanks Charl.

Solution 2

What we can also do is issue a focus gained event for the password field which will be actualised once the event queue gets a chance.

```
//: LoginDialog3.java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class LoginDialog3 extends JDialog {
  private final JTextField userName = new JTextField(8);
  private final JPasswordField password = new JPasswordField(8);
  public LoginDialog3(Frame owner) {
    super(owner, "Login Dialog", true);
    getContentPane().setLayout(new GridLayout(0,2,5,5));
    getContentPane().add(new JLabel("Username:"));
    getContentPane().add(userName);
    getContentPane().add(new JLabel("Password:"));
    getContentPane().add(password);
    pack();
    Windows.centerOnScreen(this);
    changeFocus(userName, password);
    show();
  }
  private void changeFocus(final Component source,
      final Component target) {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        target.dispatchEvent(
          new FocusEvent(source, FocusEvent.FOCUS_GAINED));
      }
    });
  }
  public String getUserName() { return userName.getText(); }
  public String getPassword() { return password.getText(); }
  public static void main(String[] args) {
    JFrame owner = new JFrame("Login Dialog");
    owner.setLocation(-1000, -1000);
    owner.show();
    owner.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    new LoginDialog3(owner);
  }
}
```

This also works perfectly, but I cannot decide which is more obscure. I suppose the 2nd solution is "better" because we can move the focus changing code out of the class into a general GUI utilities class and do this type of focus changing in a consistent way throughout the project. Also, it is probably easier with the 2nd

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

solution to hop to any component on the screen, rather than just transfer the focus to the next component.

You be the judge. Please let me know if you have a better solution to this problem, by sending email to heinz@javaspecialists.co.za.

Until next week, when I will look at what happens when you send GUI components over the network, ideas sponsored by Niko Brummer.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 013a] - Serializing GUI Components Across Network

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 13th issue of "The Java(tm) Specialists' Newsletter". Let's hope my luck does not run out in your estimation and that you don't assign my to a Java loony bin, especially with the things I do in this newsletter.

Please forward this free advanced Java newsletter to as many people as you know who might be interested in Java.

Serializing GUI Components Across Network

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

When Swing came out, I was puzzled by the following warning in the javadocs:

Warning: Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. A future release of Swing will provide support for long term persistence.

My thoughts were: No one in their right mind would want to serialize GUI components, so why are they serializable in the first place? Luckily for me, some of my friends are not in their right mind :) Many thanks to twin-dad-to-be Niko Brummer for this extremely interesting, and probably totally useless idea, though he assures me that he is using it in a real program. Some of the code in this newsletter is from him, other parts I have added. You can easily spot his code by the absense of bugs.

There are some very interesting gotchas that occur because of the dear Swing event dispatch thread, who, having failed at all attempts to find a mate, is still single, and will be for the forseeable future. Due to his hermit nature, we have to be very careful that we do not read from the GUI components or write to them from any thread except the GUI thread, or else face the wrath of a thousand deadlocks (see first newsletter in this series).

If you are serializing the component in response to pushing a GUI button, it will work, because then you are using the event dispatch thread. If you are doing it from any other thread, you may get problems (in Niko's case the event dispatch thread crashed). The solution is to wrap the gui component in an object of which you control the serialization by specifying your own writeObject() method.

I have written a ComponentSerializer class which wraps the read write functionality and you can either write a Component to an OutputStream or read it from an InputStream without having to worry about what thread you are currently in.

```
//: ComponentSerializer.java
import java.io.*;
import java.awt.*;
import javax.swing.*;
import java.lang.reflect.*; // wouldn't be right for me to send
          // you a newsletter that doesn't use reflection :)

public class ComponentSerializer {
  public void write(Component comp, OutputStream out)
      throws IOException {
    System.out.println("writing " + comp);
    ObjectOutputStream oout = new ObjectOutputStream(out);
    oout.writeObject(new ComponentEncapsulator(comp));
    oout.reset();
    oout.flush();
  }
  public Component read(InputStream in)
      throws IOException, ClassNotFoundException {
    System.out.println("reading component");
    ObjectInputStream oin = new ObjectInputStream(in);
    ComponentEncapsulator enc =
      (ComponentEncapsulator)oin.readObject();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    return enc.getComponent();
  }
  private class ComponentEncapsulator implements Serializable {
    private final Component comp;
    public ComponentEncapsulator(Component comp) {
      this.comp = comp;
    }
    public Component getComponent() {
      return comp;
    }
    private IOException defaultWriteException;
    private void writeObject(final ObjectOutputStream out)
        throws IOException {
      if (SwingUtilities.isEventDispatchThread()) {
        // This is all that is necessary if we are already in
        // the event dispatch thread, e.g. a user clicked a
        // button which caused the object to be serialized
        out.defaultWriteObject();
      } else {
        try {
          // we want to wait until the object has been written
          // before continuing.  If we called this from the
          // event dispatch thread we would get an exception
          SwingUtilities.invokeAndWait(new Runnable() {
            public void run() {
              try {
                // easiest way to indicate to the enclosing class
                // that an exception occurred is to have a member
                // which keeps the IOException
                defaultWriteException = null;
                // we call the actual write object method
                out.defaultWriteObject();
              } catch(IOException ex) {
                // oops, an exception occurred, remember the
                // exception object
                defaultWriteException = ex;
              }
            }
          });
          if (defaultWriteException != null) {
            // an exception occurred in the code above, throw it!
            throw defaultWriteException;
          }
        } catch(InterruptedException ex) {
          // I'm not quite sure what do here, perhaps:
          Thread.currentThread().interrupt();
          return;
        } catch(InvocationTargetException ex) {
          // This can actually only be a RuntimeException or an
          // Error - in either case we want to rethrow them
          Throwable target = ex.getTargetException();
          if (target instanceof RuntimeException) {
            throw (RuntimeException)target;
          } else if (target instanceof Error) {
            throw (Error)target;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
        }
        ex.printStackTrace(); // this should not happen!
        throw new RuntimeException(ex.toString());
      }
    }
  }
 }
}
```

I apologize for all the comments in the ComponentSerializer, as we all know, too many comments are often indicative of poorly written code, but I cannot think of a simpler, yet correct, way of doing this. This ComponentSerializer class should handle just about any java.awt.Component derivative thrown at it from any thread, except for components which reference non-serializable components.

We can then write a GUIServer, which accepts a ComponentEncapsulator via TCP/IP and constructs a JFrame containing the component within the ComponentEncapsulator. We only cater for one component per socket, but that could easily be changed.

```
//: GUIServer.java
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.net.*;

public class GUIServer {
  public static final int PORT = 4123;
  private static final ComponentSerializer compser =
    new ComponentSerializer();
  public GUIServer() throws IOException {
    System.out.println("Super-duper GUI SERVER started");
    ServerSocket ss = new ServerSocket(PORT);
    while(true) {
      Socket socket = ss.accept();
      try {
        JFrame frame = new JFrame(
          "Component received from " + socket);
        Component comp = compser.read(socket.getInputStream());
        frame.getContentPane().add(comp);
        frame.pack();
        frame.show();
      } catch(IOException ex) {
        ex.printStackTrace();
      } catch(ClassNotFoundException ex) {
        ex.printStackTrace();
      }
    }
  }
  public static void main(String[] args) throws IOException {
    new GUIServer();
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

We can then take a Component, for example a JScrollPane containing a JTable, and send it to any OutputStream, e.g. to the network, to the file system for short-term storage, etc. The warning in the Swing source essentially tells us we should not write GUI components onto DAT tapes for long-term archiving.

```
//: GUIExample.java
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.net.*;

public class GUIExample extends JFrame {
  public static final int PORT = 4123;
  private static final ComponentSerializer compser =
    new ComponentSerializer();
  private JScrollPane scrollPane;
  public GUIExample() {
    super("GUIExample Frame");
    scrollPane = new JScrollPane(new JTable(3,4));
    getContentPane().add(scrollPane);
    getContentPane().add(new JButton(
      new AbstractAction("Serialize Table") {
        public void actionPerformed(ActionEvent e) {
          System.out.println("Now we serialize synchronously");
          try {
            Socket socket = new Socket("localhost", PORT);
            compser.write(scrollPane, socket.getOutputStream());
            socket.close();
          } catch(IOException ex) {
            ex.printStackTrace();
          }
        }
      }), BorderLayout.SOUTH);
    setSize(400, 200);
    show();
  }
  public static void main(String[] args) throws Exception {
    GUIExample ex = new GUIExample();
    ex.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```

Please try this out - I was quite amazed that it actually worked! Start the GUIServer class, then start the GUIExample and press the button. The GUIServer should now open up another JFrame containing the JTable. Now edit the original JTable (press enter after editing, otherwise you'll get an exception when you try to serialize the table), and click on the button again. The GUIServer should now open up yet another JFrame with the JTable containing the latest values. Though I cannot think of a good application for this at the moment, I think it's quite neat that you can do that.

I've also written a second example to show you how to do an asynchronous serialization of components via another thread.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
//: GUIExample2
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.net.*;

public class GUIExample2 extends JFrame {
  public static final int PORT = 4123;
  private static final ComponentSerializer compser =
    new ComponentSerializer();
  private JPanel personalData;
  public GUIExample2() {
    super("Asynchronous Sending GUIExample2 Frame");
    personalData = new JPanel(new GridLayout(0, 2, 5, 5));
    personalData.add(new JLabel("Name: "));
    personalData.add(new JTextField());
    personalData.add(new JLabel("Age: "));
    personalData.add(new JTextField());
    getContentPane().add(personalData, BorderLayout.NORTH);
    getContentPane().add(new JButton(
      new AbstractAction("Serialize Personal Data") {
        public void actionPerformed(ActionEvent e) {
          asyncSerialize(personalData);
        }
      }), BorderLayout.SOUTH);
    setSize(400, 200);
    show();
  }
  private void asyncSerialize(final Component comp) {
    new Thread() { {start();} // start from initializer block
      public void run() {
        try {
          Socket socket = new Socket("localhost", PORT);
          compser.write(comp, socket.getOutputStream());
          socket.close();
        } catch(IOException ex) {
          ex.printStackTrace();
        }
      }
    };
  }
  public static void main(String[] args) throws Exception {
    GUIExample2 ex = new GUIExample2();
    ex.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```

When you try this out, you'll notice that the entire JPanel gets sent across the network to the server.

Were I to use this in a production environment framework, I would add the option of writing asynchronously to the ComponentSerializer and probably do the writing via a ThreadPool.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Next week I will demonstrate that the following can be **true**:

"hi there".equals(**"cheers !"**)

Seeing is believing, so until next week

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 013b] - Follow-up

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Hi again,

Imagine my horror this morning when I tried to run the code from last nights newsletter and discovered that it generated an exception! On frantic searching I figured that the JBuilder 3.0 compiler produces a different result to the SUN JDK 1.3 compiler. I had only run the program from within JBuilder, so I got quite a surprise that it did not work in SUN. The reason I'm blaming the compiler is because when I ran the class files generated by JBuilder 3.0 with the JDK 1.3 VM it works perfectly.

The problem is that if you call out.defaultWriteObject(), the two compilers have different ideas of which object you are calling this from, because you are inside an

inner class. The SUN compiler thinks you are calling it from ComponentSerializer and that is not serializable.

The solution is quite simple, just move the ComponentEncapsulator class out of the ComponentSerializer class and it should work with the JDK 1.3 compiler.

We thus end up with ComponentSerializer:

```
//: ComponentSerializer.java
import java.io.*;
import java.awt.*;
public class ComponentSerializer {
  public void write(Component comp, OutputStream out)
      throws IOException {
    System.out.println("writing " + comp);
    ObjectOutputStream oout = new ObjectOutputStream(out);
    oout.writeObject(new ComponentEncapsulator(comp));
    oout.reset();
    oout.flush();
  }
  public Component read(InputStream in)
      throws IOException, ClassNotFoundException {
    System.out.println("reading component");
    ObjectInputStream oin = new ObjectInputStream(in);
    ComponentEncapsulator enc =
      (ComponentEncapsulator)oin.readObject();
    return enc.getComponent();
  }
}
```

and ComponentEncapsulator:

```
//: ComponentEncapsulator.java
import java.io.*;
import java.awt.*;
import javax.swing.*;
import java.lang.reflect.*; // wouldn't be right for me to send
        // you a newsletter that doesn't use reflection :)
class ComponentEncapsulator implements Serializable {
  private final Component comp;
  private IOException defaultWriteException;
  public ComponentEncapsulator(Component comp) {
    this.comp = comp;
  }
  public Component getComponent() {
    return comp;
  }
  private void writeObject(final ObjectOutputStream out)
      throws IOException {
    if(SwingUtilities.isEventDispatchThread()) {
      // This is all that is necessary if we are already in
      // the event dispatch thread, e.g. a user clicked a
      // button which caused the object to be written
      out.defaultWriteObject();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    } else {
      try {
        // we want to wait until the object has been written
        // before continuing.  If we called this from the
        // event dispatch thread we would get an exception
        SwingUtilities.invokeAndWait(new Runnable() {
          public void run() {
            try {
              // easiest way to indicate to the enclosing class
              // that an exception occurred is to have a member
              // which keeps the IOException
              defaultWriteException = null;
              // we call the actual write object method
              out.defaultWriteObject();
            } catch(IOException ex) {
              // oops, an exception occurred, remember the
              // exception object
              defaultWriteException = ex;
            }
          }
        });
        if (defaultWriteException != null) {
          // an exception occurred in the code above, throw it!
          throw defaultWriteException;
        }
      } catch(InterruptedException ex) {
        // I'm not quite sure what do here, perhaps:
        Thread.currentThread().interrupt();
        return;
      } catch(InvocationTargetException ex) {
        // This can actually only be a RuntimeException or an
        // Error - in either case we want to rethrow them
        Throwable target = ex.getTargetException();
        if (target instanceof RuntimeException) {
          throw (RuntimeException)target;
        } else if (target instanceof Error) {
          throw (Error)target;
        }
        ex.printStackTrace(); // this should not happen!
        throw new RuntimeException(ex.toString());
      }
    }
  }
}
```

This highlights again that we have to be quite careful with Java. In the big project that we work on, we have standardized on the JDK 1.3 compiler and ALL our classes have to be compiled with that. This happened only after much nailbiting because of slight compiler differences of various IDEs.

Copyright 2000-2003 Maximum Solutions, South Africa

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 014] - Insane Strings

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 14th issue of "The Java(tm) Specialists' Newsletter", where we look at things that other newsletters would not dare to mention. Please please use the ideas presented in this newsletter with caution, they can really mess up a project if used (incorrectly). A lot of them are educational rather than practical so that we can understand Java better. This is true especially for this newsletter, in which we mutate Strings.

Thanks if you forwarded last week's newsletter to some of your colleagues and friends, I got a few subscriptions as a result of referrals. The membership has been growing steadily since I started this newsletter in November 2000. It has now broken through the 400 barrier and I'm thinking of moving it to a proper list server before reaching 500. My main concern is that if it is a free list server it might start sending you spam, which I really would not like. Any suggestions of free list servers that do not generate spam would be most welcome. My current preference is for Topica - which I will choose unless I hear from you.

Playing with your sanity - Strings

Have a look at the following code:

```
public class MindWarp {
  public static void main(String[] args) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    System.out.println(
      "Romeo, Romeo, wherefore art thou oh Romero?");
  }
  private static final String OH_ROMEO =
    "Romeo, Romeo, wherefore art thou oh Romero?";
  private static final Warper warper = new Warper();
}
```

If we are told that the class Warper does not produce any visible output when you construct it, what is the output of this program? The most correct answer is, "you don't know, depends on what Warper does". Now THERE's a nice question for the Sun Certified Java Programmer Examination.

In my case, running "java MindWarp" produces the following output

```
C:> java MindWarp <ENTER>
Stop this romance nonsense, or I'll be sick
```

And here is the code for Warper:

```
import java.lang.reflect.*;

public class Warper {
  private static Field stringValue;
  static {
    // String has a private char [] called "value"
    // if it does not, find the char [] and assign it to value
    try {
      stringValue = String.class.getDeclaredField("value");
    } catch(NoSuchFieldException ex) {
      // safety net in case we are running on a VM with a
      // different name for the char array.
      Field[] all = String.class.getDeclaredFields();
      for (int i=0; stringValue == null && i<all.length; i++) {
        if (all[i].getType().equals(char[].class)) {
          stringValue = all[i];
        }
      }
    }
    if (stringValue != null) {
      stringValue.setAccessible(true); // make field public
    }
  }
  public Warper() {
    try {
      stringValue.set(
        "Romeo, Romeo, wherefore art thou oh Romero?",
        "Stop this romance nonsense, or I'll be sick".
          toCharArray());
      stringValue.set("hi there", "cheers !".toCharArray());
    } catch(IllegalAccessException ex) {} // shhh
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

How is this possible? How can String manipulation in a completely different part of the program affect our class MindWarp?

To understand that, we have to look under the hood of Java. In the language specification it says in §3.10.5:

"Each string literal is a reference (§4.3) to an instance (§4.3.1, §12.5) of class String (§4.3.3). String objects have a constant value. String literals-or, more generally, strings that are the values of constant expressions (§15.28)-are "interned" so as to share unique instances, using the method String.intern."

The usefulness of this is quite obvious, we will use less memory if we have two Strings which are equivalent pointing at the same object. We can also manually intern Strings by calling the intern() method.

The language spec goes a bit further:

Literal strings within the same class (§8) in the same package (§7) represent references to the same String object (§4.3.1).
Literal strings within different classes in the same package represent references to the same String object.
Literal strings within different classes in different packages likewise represent references to the same String object.
Strings computed by constant expressions (§15.28) are computed at compile time and then treated as if they were literals.
Strings computed at run time are newly created and therefore distinct.
The result of explicitly interning a computed string is the same string as any pre-existing literal string with the same contents.

This means that if a class in another package "fiddles" with an interned String, it can cause havoc in your program. Is this a good thing? (You don't need to answer ;-)

Consider this example

```
public class StringEquals {
public static void main(String[] args) {
  System.out.println("hi there".equals("cheers !"));
}
private static final String greeting =  "hi there";
private static final Warper warper =  new Warper();
}
```

Running this against the Warper produces a result of true, which is really weird, and in my opinion, quite mind-bending. Hey, you can SEE the values there right in front of you and they are clearly NOT equal!

BTW, for simplicity, the Strings in my examples are exactly the same length, but you can change the length quite easily as well.

Last example concerns the HashCode of String, which is now cached for performance reasons mentioned in "Java Idiom and Performance Guide", ISBN 0130142603. (Just for the record, I was never and am still not convinced that caching the String hash

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

code in a wrapper object is a good idea, but caching it in String itself is almost acceptable, considering String literals.)

```
public class CachingHashcode {
  public static void main(String[] args) {
    java.util.Map map = new java.util.HashMap();
    map.put("hi there", "You found the value");
     new Warper();
    System.out.println(map.get("hi there"));
    System.out.println(map);
  }
  private static final String greeting = "hi there";
}
```

The output under JDK 1.3 is:

```
You found the value
{cheers != You found the value}
```

Under JDK 1.2 it is

```
null
{cheers != You found the value}
```

This is because in the JDK 1.3 SUN is caching the hash code so if it once calculated, it doesn't get recalculated, so if the value field changes, the hashcode stays the same.

Imagine trying to debug this program where SOMEWHERE, one of your hackers has done a "workaround" by modifying a String literal. The thought scares me.

There is of course a small keyword that would have stopped this problem, namely "final". I got into the habit a few months ago to make all my data members final where possible, and it has paid off more than once. Surprisingly, the char array in String is not final.

Consider the following example code:

```
public class Bla {
  private char[] c1 = "hello".toCharArray();
  private final char[] c2 = "bye".toCharArray();
  public String toString() {
    return c1 + ", " + c2;
  }
}

import java.lang.reflect.*;

public class BlaTest {
  private static Field c1;
  private static Field c2;
  static {
    try {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      c1 = Bla.class.getDeclaredField("c1");
      c1.setAccessible(true);
      c2 = Bla.class.getDeclaredField("c2");
      c2.setAccessible(true);
    } catch(NoSuchFieldException ex) { }
  }
  public static void main(String[] args) {
    Bla bla = new Bla();
    try {
      c1.set(bla, "mutatedc1".toCharArray());
      c2.set(bla, "mutatedc2".toCharArray());
    } catch(IllegalAccessException ex) {
      ex.printStackTrace();
    }
    System.out.println(bla);
  }
}
```

When I run my program, I can quite happily change c1, but when I try to change c2 I get an exception. String has no reason for value to be non-final, so it should be final. If you have contacts at SUN, please forward them this newsletter and ask them to make value final. It might stop some nasty Java viruses from completely messing up the JVM.

Until next week, and please remember to forward this newsletter and send me your comments.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 015] - Implementing a SoftReference based HashMap

Author: Dr. Heinz M. Kabutz

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 15th issue of "The Java(tm) Specialists' Newsletter", this week we are looking at an extremely useful addition to the Java language, namely soft and weak references. In a nutshell, the main difference between the two seems to be that the SoftReference has some notion of remembering when last it was accessed, which may be used by the GC (if it feels like it) to release little-used SoftReferences first.

Next week I will be doing some training in Germany, so I will not be able to send out a newsletter, as I have not decided on a list server yet. Hopefully the week after that you'll have your normal fix again. Please continue forwarding this newsletter to your local JUG, friends and family who are interested in Java. If you do send the newsletter to a closed user list, kindly tell me, I like having an indication of how many souls are reading my newsletter. Similarly, if you quote from my newsletter, be so kind as to attribute the source.

Many thanks to Sydney Redelinghuys for his input and corrections in this newsletter. (Sydney wants to spend a bit of time working in London, so if you are in the London area and would like the services of one of the most brilliant Java programmers I know, please send Sydney an email.

Once again I have seen the value in actually testing my code, as there was a serious error in my code which I only found through my "unit" test.

Implementing a SoftReference based HashMap

Java is slow. Java is a memory hog.

But, if I have to write a network application, I would not hesitate for a second to use Java. Why? Because Java shines in threaded, network-based applications and because over the years, I have recognised the weaknesses of Java and have learnt (the hard way) what I should do, or not do, to avoid running into too serious problems. Recognising the problems takes a while to learn, which is why most Java jobs require you to have at least 2 years of Java programming experience.

One of the most common places of memory wastage is in hash maps, so SUN have provided a WeakHashMap to minimize memory usage in caches implemented using maps. A WeakHashMap stores the keys using WeakReference objects, which means in layman's terms that as soon as the key is not referenced from somewhere else in your program, the entry may be removed and is available for garbage collection. (Have a look at the JavaDocs for the java.util.WeakHashMap and java.lang.ref.WeakReference for more information) Say you write a middle tier that provides an OO view of your database via an application server. What you really want is for the values to be automatically released, rather than the keys. If you put all the objects into a normal HashMap, you can easily run out of memory when you access many different objects from different clients. But if you store the objects in a WeakHashMap they are cleared as soon as your clients is not referencing them

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

anymore. What you really want, however, is to only have the objects released when the VM is running low on memory, since that is when you get problems.

Enter SoftReferences. As far as I understand, a SoftReference will only be garbage collected when the VM is running low on memory and the object it is pointing to is not accessed from a normal (hard) reference. This is probably a better option to use for the HashMap values than a WeakReference, but the default JDK collections don't include a GC-friendly HashMap based on values and neither does it provide a SoftReference based HashMap.

Before I show you a SoftHashMap implementation, based on ideas by Sydney (what's up doc?) Redelinghuys, I need to explain some ideas which will make our SoftHashMap more optimal.

Each time we change the map (put, remove, clear) or ask for its size, we first want to go through the map and throw away entries for which the values have been garbage collected. It is quite easy to find out which soft references have been cleared. You can give the SoftReference a ReferenceQueue to which it is added when it is garbage collected.
We don't want our hash map to be bullied by the garbage collector, so we provide the option of the map itself keeping a hard link to the last couple of objects (typically 100).
The SoftHashMap will use a variant of the Decorator pattern to add this functionality to an internally kept java.util.HashMap. I'm busy working on a Design Patterns course based on the GOF book, let me know if you want further information.

Without further ado, here comes the SoftHashMap:

```
//: SoftHashMap.java
import java.util.*;
import java.lang.ref.*;

public class SoftHashMap extends AbstractMap {
  /** The internal HashMap that will hold the SoftReference. */
  private final Map hash = new HashMap();
  /** The number of "hard" references to hold internally. */
  private final int HARD_SIZE;
  /** The FIFO list of hard references, order of last access. */
  private final LinkedList hardCache = new LinkedList();
  /** Reference queue for cleared SoftReference objects. */
  private final ReferenceQueue queue = new ReferenceQueue();

  public SoftHashMap() { this(100); }
  public SoftHashMap(int hardSize) { HARD_SIZE = hardSize; }

  public Object get(Object key) {
    Object result = null;
    // We get the SoftReference represented by that key
    SoftReference soft_ref = (SoftReference)hash.get(key);
    if (soft_ref != null) {
      // From the SoftReference we get the value, which can be
      // null if it was not in the map, or it was removed in
      // the processQueue() method defined below
      result = soft_ref.get();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    if (result == null) {
      // If the value has been garbage collected, remove the
      // entry from the HashMap.
      hash.remove(key);
    } else {
      // We now add this object to the beginning of the hard
      // reference queue.  One reference can occur more than
      // once, because lookups of the FIFO queue are slow, so
      // we don't want to search through it each time to remove
      // duplicates.
      hardCache.addFirst(result);
      if (hardCache.size() > HARD_SIZE) {
        // Remove the last entry if list longer than HARD_SIZE
        hardCache.removeLast();
      }
    }
  }
  return result;
}

/** We define our own subclass of SoftReference which contains
 not only the value but also the key to make it easier to find
 the entry in the HashMap after it's been garbage collected. */
private static class SoftValue extends SoftReference {
  private final Object key; // always make data member final
  /** Did you know that an outer class can access private data
   members and methods of an inner class?  I didn't know that!
   I thought it was only the inner class who could access the
   outer class's private information.  An outer class can also
   access private members of an inner class inside its inner
   class. */
  private SoftValue(Object k, Object key, ReferenceQueue q) {
    super(k, q);
    this.key = key;
  }
}

/** Here we go through the ReferenceQueue and remove garbage
 collected SoftValue objects from the HashMap by looking them
 up using the SoftValue.key data member. */
private void processQueue() {
  SoftValue sv;
  while ((sv = (SoftValue)queue.poll()) != null) {
    hash.remove(sv.key); // we can access private data!
  }
}
/** Here we put the key, value pair into the HashMap using
 a SoftValue object. */
public Object put(Object key, Object value) {
  processQueue(); // throw out garbage collected values first
  return hash.put(key, new SoftValue(value, key, queue));
}
public Object remove(Object key) {
  processQueue(); // throw out garbage collected values first
  return hash.remove(key);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }
  public void clear() {
    hardCache.clear();
    processQueue(); // throw out garbage collected values
    hash.clear();
  }
  public int size() {
    processQueue(); // throw out garbage collected values first
    return hash.size();
  }
  public Set entrySet() {
    // no, no, you may NOT do that!!! GRRR
    throw new UnsupportedOperationException();
  }
}
```

And here comes some test code that demonstrates to a certain degree that I'm not talking complete bullsh*t. Soft and weak references are quite difficult to experiment with as there is a lot of freedom left to the writers of the JVM of how they must implement them. I wish the implementation would hold back longer before removing these references, that the JVM would wait until it is really running low on memory before clearing them, but unfortunately I am not the one who wrote the JVM. I have tried to question the authors of the java.lang.ref package to find out what the strategy is for references in future versions, but have not had any response yet.

```
//: SoftHashMapTest.java
import java.lang.ref.*;
import java.util.*;

public class SoftHashMapTest {
  private static void print(Map map) {
    System.out.println("One=" + map.get("One"));
    System.out.println("Two=" + map.get("Two"));
    System.out.println("Three=" + map.get("Three"));
    System.out.println("Four=" + map.get("Four"));
    System.out.println("Five=" + map.get("Five"));
  }
  private static void testMap(Map map) {
    System.out.println("Testing " + map.getClass());
    map.put("One", new Integer(1));
    map.put("Two", new Integer(2));
    map.put("Three", new Integer(3));
    map.put("Four", new Integer(4));
    map.put("Five", new Integer(5));
    print(map);
    byte[] block = new byte[10*1024*1024]; // 10 MB
    print(map);
  }
  public static void main(String[] args) {
    testMap(new HashMap());
    testMap(new SoftHashMap(2));
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Until next week, and please remember to forward this newsletter and send me your comments.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 016] - Blocking Queue

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 16th issue of "The Java(tm) Specialists' Newsletter", written in a dreary-weathered-Germany. Since I'm a summer person, I really like living in South Africa where we have 9 months of summer and 3 months of sort-of-winter. It's quite difficult to explain to my 2-year old son the concepts of snow, snow-man, snow-ball, etc. Probably as difficult as explaining to a German child the concepts of cloudless-sky, beach, BSE-free meat, etc.

Next week I will again not be able to post the newsletter due to international travelling (Mauritius), but the week after that I will demonstrate how it is possible to write type-safe enum types in Java using inner classes and how it is possible to "switch" on their object references. Switch statements should never be used, but it is nevertheless fascinating to watch how the Java language constructs can be abused...

Blocking Queues for inter-thread communication

This week I want to speak about a very useful construct that we use for inter-thread communication, called a blocking queue. Quite often in threaded applications we

have a producer-consumer situation where some threads want to pop jobs onto a queue, and some other worker threads want to remove jobs from the queue and then execute them. It is quite useful in such circumstances to write a queue which blocks on pop when there is nothing on the queue. Otherwise the consumers would have to poll, and polling is not very good because it wastes CPU cycles.

I have written a very simple version of the BlockingQueue, a more advanced version would include alarms that are generated when the queue reaches a certain length.

Warning Advanced:
When I write pieces of code which are synchronized, I usually avoid synchronizing on "this" or marking the whole method as synchronized. When you synchronize on "this" inside the class, it might happen that other code outside of your control also synchronize on the handle to your object, or worse, call notify on your handle. This would severely mess up your well-written BlockingQueue code. I therefore as a habit always use private data members as locks inside a class, in this case I use the private queue data member.

Another disadvantage of indiscriminately synchronizing on "this" is that it is very easy to then lock out parts of your class which do not necessarily have to be locked out from each other. For example, I might have a list of listeners in my BlockingQueue which are notified when the list gets too long. Adding and removing such listeners from the BlockingQueue should be synchronized, but you do not have to synchronize in respect of the push and pop operations, otherwise you limit concurrency.

```
//: BlockingQueue.java
import java.util.*;
public class BlockingQueue {
  /**
    It makes logical sense to use a linked list for a FIFO queue,
    although an ArrayList is usually more efficient for a short
    queue (on most VMs).
   */
  private final LinkedList queue =  new LinkedList();
  /**
    This method pushes an object onto the end of the queue, and
    then notifies one of the waiting threads.
   */
  public void push(Object o) {
    synchronized(queue) {
      queue.add(o);
      queue.notify();
    }
  }
  /**
    The pop operation blocks until either an object is returned
    or the thread is interrupted, in which case it throws an
    InterruptedException.
   */
  public Object pop()  throws InterruptedException {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    synchronized(queue) {
      while (queue.isEmpty()) {
        queue.wait();
      }
      return queue.removeFirst();
    }
  }
  /** Return the number of elements currently in the queue. */
  public int size() {
    return queue.size();
  }
}
```

Now we've got a nice little test case that uses the blocking queue for 10 worker threads which will each pull as many tasks as possible from the queue. To end the test, we put one poison pill onto the queue for each of the worker threads, which, when executed, interrupts the current thread (evil laughter).

```
//: BlockingQueueTest.java
public class BlockingQueueTest {
  private final BlockingQueue bq = new BlockingQueue();
  /**
    The Worker thread is not very robust.  If a RuntimeException
    occurse in the run method, the thread will stop.
   */
  private class Worker extends Thread {
    public Worker(String name) { super(name); start(); }
    public void run() {
      try {
        while(!isInterrupted()) {
          ((Runnable)bq.pop()).run();
        }
      } catch(InterruptedException ex) {}
      System.out.println(getName() + " finished");
    }
  }
  public BlockingQueueTest() {
    // We create 10 threads as workers
    Thread[] workers = new Thread[10];
    for (int i=0; i<workers.length; i++)
      workers[i] = new Worker("Worker Thread " + i);
    // We then push 100 commands onto the queue
    for (int i=0; i<100; i++) {
      final String msg = "Task " + i + " completed";
      bq.push(new Runnable() {
        public void run() {
          System.out.println(msg);
          // Sleep a random amount of time, up to 1 second
          try { Thread.sleep((long)(Math.random()* 1000)); }
          catch(InterruptedException ex) { }
        }
      });
    }
    // We then push one "poison pill" onto the queue for each
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    // worker thread, which will only be processed once the other
    // tasks are completed.
    for (int i=0; i<workers.length; i++) {
      bq.push(new Runnable() {
        public void run() {
          Thread.currentThread().interrupt();
        }
      });
    }
    // Lastly we join ourself to each of the Worker threads, so
    // that we only continue once all the worker threads are
    // finished.
    for (int i=0; i<workers.length; i++) {
      try {
        workers[i].join();
      } catch(InterruptedException ex) {}
    }
    System.out.println("BlockingQueueTest finished");
  }
  public static void main(String[] args) throws Exception{
    new BlockingQueueTest();
  }
}
```

The concepts in the newsletter can be expanded quite a bit. They could, for example, be used as a basis for implementing a ThreadPool, or otherwise you can implement an "ActiveQueue" which performs callbacks to listeners each time an event is pushed onto the queue via a Thread running inside the ActiveQueue.

It is also possible to use PipedInputStream and PipedOutputStream to send messages between threads, but then you have to set up a whole protocol, and if you want to exchange objects you have to use ObjectOutputStream which will be alot slower than just passing handles.

Until next week, and please remember to forward this newsletter in its entirety to as many Java users as you know.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

..... [Issue 017a] - Switching on Object Handles

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 17th issue of "The Java(tm) Specialists' Newsletter", after a few very busy days in the Mauritian Paradise. Mauritius is a wonderful place to go to, with extremely friendly people all around, treating you like kings even in business. It's definitely worth a vacation, I wish I had gotten one while I was there ;-)

Please remember to forward this newsletter to anyone who might be interested, friends and foe.

Simulating Switch Statements on Handles with try-catch

This week I will talk about a really completely useless idea, how to use switch/case statements on type-safe constants in Java. This idea occurred to me while I was talking to a bunch of programmers about TCP/IP programming, I don't think my topic of conversation had anything to do with the hare-brained idea presented in this newsletter. My listeners saw my eyes take on a distant gaze and I muttered "hmmm, I wonder if..." so here you go.

First I need to bore you with a monologe of why switch statements are bad and why you should never use them. Switch statements herald from a time before we used Object Orientation, Encapsulation and Polymorphism, and were mostly used to write methods which would do different things depending on the type of record we had passed the method. For example, say we had a CAD system, with a triangle, rectangle and circle, we could say:

```
public interface Constants {
  int TRIANGLE_SHAPE = 0;
  int RECTANGLE_SHAPE = 1;
  int CIRCLE_SHAPE = 2;
}
```

Without encapsulation, we would then have a struct or class without methods, looking like this:

```
public class Shape {
  public int type;
  public java.awt.Color color;
}
```

We would then have a CAD system for drawing these shapes, such as

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public class CADSystem implements Constants {
  public void draw(Shape shape) {
    switch(shape.type) {
      case TRIANGLE_SHAPE: // some code which draws a triangle
        System.out.println("Triangle with color " + shape.color);
        break;
      case RECTANGLE_SHAPE: // some code which draws a rectangle
        System.out.println("Rectangle with color " +shape.color);
        break;
      case CIRCLE_SHAPE: // some code which draws a circle
        System.out.println("Circle with color " + shape.color);
        break;
      default: // error only found at runtime
        throw new IllegalArgumentException(
          "Shape has illegal type " + shape.type);
    }
  }
}
```

This was the old procedural way of writing such code. The result was code where it was extremely challenging to add new types. In addition, in Java such code is very dangerous because we don't have enumerated types and you cannot switch on object references (well, have a look further down on how you actually "can"). You therefore could not be sure at compile time if you had defined the method for all the various types in your CADSystem.

The answer in OO is to use inheritance, polymorphism and encapsulation, the above example would thus be written as:

```
public abstract class Shape {
  private final java.awt.Color color;
  protected Shape(java.awt.Color color) { this.color = color; }
  public java.awt.Color getColor() { return color; }
  public abstract void draw();
}

public class Triangle extends Shape {
  public Triangle(java.awt.Color color) { super(color); }
  public void draw() {
    System.out.println("Triangle with color " + getColor());
  }
}

public class Rectangle extends Shape {
  public Rectangle(java.awt.Color color) { super(color); }
  public void draw() {
    System.out.println("Rectangle with color " + getColor());
  }
}

public class Circle extends Shape {
  public Circle(java.awt.Color color) { super(color); }
  public void draw() {
    System.out.println("Circle with color " + getColor());
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }
}

public class CADSystem {
  public void draw(Shape shape) {
    shape.draw();
  }
}
```

Now if we forget to implement one of the draw methods, we'll immediately get a compile-time error. Of course, if we extend Rectangle and forget to implement the draw method we'll get the wrong shape, so a certain level of diligence in testing is still required.

It is possible to take a switch statements and transform it to polymorphism using various refactorings. In previous newsletters I mentioned the book "Refactoring" by Martin Fowler. In case you hadn't noticed, I am a fan (of the book, that is). In that book you can find refactorings to transform a switch/case statement to polymorphism or polymorphism back to a switch/case statement.

So, in the unfortunate case (haha, pun intended) that you want to use a switch-type of construct but you don't want to worry about the anonymity of using int's as type identifiers, how do you do it?

We demonstrate by using a TransactionType class which defines the transaction isolation levels you find in most enterprise systems. The isolation types are None, Read Uncommitted, Read Committed, Repeatable Read and Serializable. The point of this newsletter is not to describe transaction isolations, so I won't go into what they all mean. Rumour has it though, that if you use them without knowing what they mean, you will get a system which doesn't work, HA.

We define a TransactionType superclass with a private constructor, so that it is not possible to construct instances of these types or to subclass it, except from within the type. The constructor takes a name as a description, which can be returned via the toString() method. The reason why the type class has to be Throwable will become clear in the example.

We then make public static inner classes for each of the types, again with private constructors, and make public static final instances of these types in each of the inner classes. The reason why we need classes and instances will also become clearer in the example.

```
//: TransactionType.java
public class TransactionType extends Throwable {
private final String name;
private TransactionType(String name) {
  this.name = name;
}
public String toString() { return name; }
public static class None extends TransactionType {
  public static final TransactionType type = new None();
  private None() { super("None"); }
}
public static class ReadUncommitted extends TransactionType {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public static final TransactionType type =
    new ReadUncommitted();
  private ReadUncommitted() { super("ReadUncommitted"); }
}
public static class ReadCommitted extends TransactionType {
  public static final TransactionType type =
    new ReadCommitted();
  private ReadCommitted() { super("ReadCommitted"); }
}
public static class RepeatableRead extends TransactionType {
  public static final TransactionType type =
    new RepeatableRead();
  private RepeatableRead() { super("RepeatableRead"); }
}
public static class Serializable extends TransactionType {
  public static final TransactionType type =
    new Serializable();
  private Serializable() { super("Serializable"); }
}
}
```

How does such a type help us to make safe types which we can switch on? The answer is that we use a construct which is not really meant to be used as a switch, but which acts as one nevertheless, namely the throw-catch construct. We simply throw the type, which can be a handle to a TransactionType object, and the exception handling mechanism sorts out which catch to call. Yes, I can hear you all groaning now with pearls of sweat caused by fear, but this really does work. For syntactic sugar, we can import the inner classes using "import TransactionType.*" after which we can refer to the inner class simply by their name "ReadCommitted". We can of course also use the full name such as Transaction.None instead of importing the inner classes.

```
//: SwitchingOnObjects.java
import TransactionType.*;
public class SwitchingOnObjects {
  public static void switchStatement(TransactionType transact) {
    try {
      throw transact;
    } catch(TransactionType.None type) {
      System.out.println("Case None received");
    } catch(ReadUncommitted type) {
      System.out.println("Case Read Uncommitted");
    } catch(ReadCommitted type) {
      System.out.println("Case Read Committed");
    } catch(RepeatableRead type) {
      System.out.println("Case Repeatable Read");
    } catch(TransactionType type) {
      System.out.println("Default");
    }
  }
  public static void main(String[] args) {
    switchStatement(TransactionType.None.type);
    switchStatement(ReadUncommitted.type);
    switchStatement(ReadCommitted.type);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    switchStatement(RepeatableRead.type);
    switchStatement(Serializable.type);
  }
}
```

Try it out, the exception handling mechanism works quite well for this. There are a few pointers you have to follow if you want to use this:

Don't ever catch "Throwable" as the default case. You should rather catch the type base class, such as TransactionType. Otherwise you run the risk of catching RuntimeException and Error classes, such as OutOfMemoryError.
Make sure that ALL the types are public inner classes of the type base class.
Make sure that all the constructors are private.
Lastly, rather use polymorphism to achieve this effect. Switch/Case code is really messy to maintain and very error- prone.

I've already donned my asbethos suit for the criticisms from hard-nosed C/C++ programmers who think switch/case is great and from weeny Java purists who think that switch/case is completely unacceptable. Flame away...

Until next week, and please remember to forward this newsletter in its entirety to as many Java users as you know.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 017b] - Follow-up

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

As expected, I received interesting comments as a result of my last newsletter, and more is sure to follow after the long weekend.

Here some comments from someone at EnterpriseDeveloper, makers of the excellent JCertify product for practicing for the SUN Certified Java Programmer Examination:

OK abestos suit man,

I'm more concerned about the encouragement of using exceptions to modify/implement regular program flow.

I just picked up some code from another group where they were doing Exceptions to break out of Whiles, For's, thowing exceptions as a kind of break, ... Not only is it a performance hit, it makes for really "nasty" code.

Perhaps you can put a summary statement on your next issue reminding people that Exceptions are not meant to be used as Loop control / Loop Control flow enhancement. I could see someone putting your switch idea inside a BIG For loop - say a list of 1,000 items, and causing 1,000 exceptions.

Very interesting reading - I always like "pushing the envelope", I'm just somewhat concerned about the influence to those "more dangerous" with such an idea. There is an interpretation among those newer to Java that Exception Handling is good for general Program Flow enhancement (Loop breaking, break statement replacement, throwing Exceptions within a repeating loop,

Exceptions are very much abused by those coming from the C world, or non OO people coming from C++

My response:

I did some tests and found that the performance wasn't that bad, it was actually better than a list of if-instanceof-elses. Note that I am specifically not constructing a new exception, and the construction of exceptions is what takes the time, as far as I know.

Throwing exceptions to change program flow is VERY bad, I agree 100%. I am intending on sending a follow-up to yesterdays newsletter in which I warn again that you shouldn't do this. Perhaps one of the "dangerous" will say: but I've been doing that for a long time...

His response:

There is some degradation to being in a Try / Catch block, but nothing compared to the creation of an Exception. What do you mean by "Not that Bad" - less than 5% degradation (tight loop)?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

YOU may not be creating exceptions when using this, and you are probably interpreting it as "I am only condoning the use of this particular set of lines of code". It may encourage *OTHERS* to continue, or even expand, bad Exception Handling practices. They won't know about the performance cost, and they won't understand this could get "nasty" really fast by extending this concept to other programming practices (and adding REAL Exceptions).

But I still like the "pushing the envelope" aspect of that code.

In addition, I've included code for a TransactionType class that has a type id int which can be used in a switch statement. This is not as safe, but "cleaner" to use instead of abusing try. I also included a test for multiple if-instanceof-else which turned out slower, and lastly an implementation of using the strategy pattern, which was the fastest of all.

```
//: TransactionType.java
public class TransactionType extends Throwable {
  private static int counter = 0;
  private final int id = counter++;
  public final int getId() { return id; }
  private final String name;
  private TransactionType(String name) {
    this.name = name;
  }
  public String toString() { return name; }
  public static class None extends TransactionType {
    public static final TransactionType type = new None();
    private None() { super("None"); }
  }
  public static class ReadUncommitted extends TransactionType {
    public static final TransactionType type =
      new ReadUncommitted();
    private ReadUncommitted() { super("ReadUncommitted"); }
  }
  public static class ReadCommitted extends TransactionType {
    public static final TransactionType type =
      new ReadCommitted();
    private ReadCommitted() { super("ReadCommitted"); }
  }
  public static class RepeatableRead extends TransactionType {
    public static final TransactionType type =
      new RepeatableRead();
    private RepeatableRead() { super("RepeatableRead"); }
  }
  public static class Serializable extends TransactionType {
    public static final TransactionType type =
      new Serializable();
    private Serializable() { super("Serializable"); }
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Now we have a performance test which investigates the various options. Fortunately, the fastest was using Polymorphism and the Strategy design pattern, as I had hoped.

```
//: Performance.java
import TransactionType.*;
public class Performance {
  public static void switchOnObject(TransactionType transact) {
    try {
      throw transact;
    } catch(TransactionType.None type) {
    } catch(ReadUncommitted type) {
    } catch(ReadCommitted type) {
    } catch(RepeatableRead type) {
    } catch(TransactionType type) {
    }
  }
  public static void switchOnInt(TransactionType transact) {
    switch(transact.getId()) {
    case 0: break;
    case 1: break;
    case 2: break;
    case 3: break;
    case 4: break;
    default: break;
    }
  }
  public static void switchInstanceof(TransactionType transact) {
    if (transact instanceof None) {
    } else if (transact instanceof ReadUncommitted) {
    } else if (transact instanceof ReadCommitted) {
    } else if (transact instanceof RepeatableRead) {
    }
  }
  public static abstract class TransactionTypeStrategy {
    public abstract void doSomething();
  }
  public static class RepeatableReadStrategy extends
      TransactionTypeStrategy {
    public void doSomething() {}
  }
  public static void main(String[] args) {
    long time = -System.currentTimeMillis();
    for (int i=0; i<1000000; i++) {
      switchOnObject(RepeatableRead.type);
    }
    time += System.currentTimeMillis();
    System.out.println("Switching 1000000 times on objects " +
      time + "ms");

    time = -System.currentTimeMillis();
    for (int i=0; i<1000000; i++) {
      switchOnInt(RepeatableRead.type);
    }
    time += System.currentTimeMillis();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    System.out.println("Switching 1000000 times on ints " +
      time + "ms");

    time = -System.currentTimeMillis();
    for (int i=0; i<1000000; i++) {
      switchInstanceof(RepeatableRead.type);
    }
    time += System.currentTimeMillis();
    System.out.println("Switching 1000000 times using " +
      "instanceofs " + time + "ms");

    time = -System.currentTimeMillis();
    TransactionTypeStrategy strategy =
      new RepeatableReadStrategy();
    for (int i=0; i<1000000; i++) {
      strategy.doSomething();
    }
    time += System.currentTimeMillis();
    System.out.println("Switching 1000000 times using " +
      "Strategy Polymorphism pattern " + time + "ms");
  }
}
```

Result on my notebook:

Switching 1000000 times on objects 481ms
Switching 1000000 times on ints 40ms
Switching 1000000 times using instanceofs 1012ms
Switching 1000000 times using Strategy Polymorphism pattern 30ms

So, for those where it's not clear yet, please don't use exceptions for ANYTHING except error handling. Don't use them to jump out of loops, do switch statements, or find out what methods you were called from.

Incidentally, in one of my first newsletters (004), I mentioned that I would like to be able to get a stack trace without having to construct an exception. So far, no-one has come up with a solution, but I've found a way which *might* work using JNI.

Regards, until next week...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 018] - Class names don't identify a class

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 18th issue of "The Java(tm) Specialists' Newsletter", now sent to 38 countries on all continents of our globe. Please remember to forward the newsletter to others who might be interested.

BEFORE YOU FILE THIS NEWSLETTER ...

I have a question for you:

South Africa is a beautiful country with a relatively stable economy, compared to other developing countries. The South- Western part of South Africa, called the Western Cape, is an especially great place to live, mainly because of the stunning environment, weather, etc. For some strange reason, we are losing a lot of programmers to the land of mist and the land of abundance. There are those in the private sector willing to invest significant money to realise their dream of the Western Cape becoming another success story in the computer world.

Now the question: If you used to live in South Africa, what would entice you to come back to the Western Cape? If you are already living in South Africa, what would prevent you from leaving? Lastly, if you are an IT professional living outside of South Africa, what "things" would attact you to move to the Western Cape?

I'm very interested to hear your opinions on this question, so don't delay, answer today :-)

Class names don't identify a class

This week I want to introduce the concepts of having Class objects in the VM with the exact same name, but being completely different. I had the opportunity to question Dr. Jung, who has been using this for a while now, and at long last, my small brain has made "click" and I understand (I think). In two weeks time, he is going to write the newsletter (he wrote the piece on dynamic proxies) and he will demonstrate how you can build up a sibling hierarchy of class loaders which you can use to automatically redeploy classes and find dependencies, etc. I didn't quite follow all of his explanations, so I'm looking forward to read what he has to say about this

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

topic. In the meantime, to prepare us all, I've written a simple example to demonstrate how it works.

In JDK 1.2, SUN added a new approach to class loading which allows you to identify classes not only by the class name, but also by the context in which it was loaded. We can set the ClassLoader for a Thread, which we can then use to load classes. By having a different classloader, you are effectively constructing a new instance of Class, which is then completely different from other instances of Class. This was, I think, also possible in JDK 1.1, but it is much easier to make an instance of a ClassLoader in JDK 1.2 as there is now a URLClassLoader which you can point to a directory where it will load the classes from.

For example, say we have two directories, a1 and a2. In each of these directories we have a class A:

```
//: a1/A.java
public class A {
  public String toString() {  return "This is the first class"; }
}
//: a2/A.java
public class A {
  public String toString() {  return "This is the second class"; }
}
```

As you will agree, the two classes are completely different. They can have different method definitions, data members or access control. The normal way of using these two classes is to choose at compile/run time which one you wish to use. For example, we may have a class NormalTest below:

```
//: NormalTest.java
public class NormalTest {
  public static void main(String[] args) {
    System.out.println(new A());
  }
}
```

To compile this class, we have to specify the directory where A resides, either a1 or a2. Since the signature is the same, we can compile with one class and run with the other class if we want.

```
javac -classpath .;a1 NormalTest.java
java -classpath .;a2 NormalTest
```

would result in "This is the second class" being displayed on the console.

What happens if we want to have instances of both A classes in use at the same time? Normally we cannot do that, but if we use ClassLoaders we can.

```
//: Loader.java
import java.net.*;
public class Loader {
  public static void main(String[] args) throws Exception {
    ClassLoader a1 =  new URLClassLoader(
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
new URL[] {new URL("file:a1/")}, null);
   ClassLoader a2 = new URLClassLoader(
new URL[] {new URL("file:a2/")}, null);
   Class c1 = a1.loadClass("A");
   Class c2 = a2.loadClass("A");
   System.out.println("c1.toString(): " + c1);
   System.out.println("c2.toString(): " + c2);
   System.out.println("c1.equals(c2): " + c1.equals(c2));
   System.out.println("c1.newInstance(): " + c1.newInstance());
   System.out.println("c2.newInstance(): " + c2.newInstance());
  }
}
```

The two classes, both called "A", are loaded with different ClassLoader objects, and are thus, as far as the VM is concerned, different classes altogether. We can print the names of the classes, compare the two classes (even though their names are the same, the classes are not equal, you should therefore never compare just the class names if you want to compare classes, rather use the Class.equals() method), and make instances of them by calling the newInstance() method.

The output if we run Loader is:

```
c1.toString(): class A
c2.toString(): class A
c1.equals(c2): false
c1.newInstance(): This is the first class
c2.newInstance(): This is the second class
```

We can also let these two "A" classes have a common superclass. For example, say we have a superclass called Parent.java located in the root directory:

```
//: Parent.java
public class Parent {
  public String toString() {
    return "Thanks for caring... but what do you want??? ";
  }
}
```

And our A.java classes are now written as:

```
//: a1/A.java
public class A extends Parent {
  public String toString() {
    return super.toString() + "This is the first class";
  }
}
//: a2/A.java
public class A extends Parent {
  public String toString() {
    return super.toString() + "This is the second class";
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

We then need to have a common parent ClassLoader which we use to load the Parent.class file. We have to specify the location to start looking for Parent.class, and the locations are searched in a hierarchical fasion. Note that the compile-time Parent class is loaded with a different ClassLoader to the one loaded with the URLClassLoader called "parent" so they refer to a different class altogether, which means we cannot type-cast an instance of "A" to a Parent. Also, if we load the class "Parent" without using the classloader, we will see that it is not equal to the superclass of "c1".

```
//: Loader.java
import java.net.*;
public class Loader {
  public static void main(String[] args) throws Exception {
    ClassLoader parent = new URLClassLoader(
new URL[] {new URL("file:./")}, null);
    ClassLoader a1 = new URLClassLoader(
new URL[] {new URL("file:a1/")}, parent);
    ClassLoader a2 = new URLClassLoader(
new URL[] {new URL("file:a2/")}, parent);
    Class c1 = a1.loadClass("A");
    Class c2 = a2.loadClass("A");
    System.out.println(c1.newInstance());
    System.out.println(c2.newInstance());
    System.out.println(
c1.getSuperclass().equals(c2.getSuperclass()));
    System.out.println(
Class.forName("Parent").equals(c1.getSuperclass()));
    try {
Parent p = (Parent)c1.newInstance();
    } catch(ClassCastException ex) {
ex.printStackTrace();
    }
    System.out.println("We expected to get ClassCastException");
  }
}
Thanks for caring... but what do you want??? This is the first class
Thanks for caring... but what do you want??? This is the second class
true
false
java.lang.ClassCastException: A
        at Loader.main(Loader.java:18)
We expected to get ClassCastException
```

Note that the super classes of both "A" classes are equal.

Where is this all this useful? It is very useful when you have an application server into which you want to deploy business objects written by different people. It is entirely feasible that you have two developers with different versions of classes deploying their applications onto the same server. You don't necessarily want to start a new VM for each deployment, and so with ClassLoaders it is possible to have lots of classes with the same name running in the same memory space but not conflicting with one another. They would also share the common JDK classes with one another, so we would not have to have a java.util.ArrayList class loaded for each of the ClassLoaders.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Until next week, and please remember to forward this newsletter in its entirety to as many Java users as you know who would be interested.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 019] - Finding Lost Frames

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 19th issue of "The Java(tm) Specialists' Newsletter". This newsletter is my contribution to advancing Java skills, and is meant as an informative, advanced newsletter that hopefully will tell you things you won't find in your average newsletter. Please let me know when my content becomes too ordinary or my style too stolid.

Thanks to all those of you who responded to last week's question, there were some interesting opinions flying around, seems like the real reasons why people go/come/stay in places is because of relationships and friendships, with money, technology, etc. only being secondary. With apologies to my readers from the erstwhile Soviet Union, it seems that was quite a well-known fact to the Politburo, but perhaps I've read too many spy stories.

Finding Lost Frames

Something that I have encountered in some almost-complete projects, were dialogs that were not bound to frames. When you construct a dialog, you should specify who

the owner is, which is either a frame or another dialog. It is, however, also possible to specify "null" as the owner, which causes very irritating problems. In MS Windows, if a modal dialog does not have an owner, and you have moved away from it, for example if you quickly switched to Outlook while waiting for it to start up, the only way to get back to the dialog is by using ALT+Tab. If you click on the frame icon on the toolbar, you will get just the frame, not the dialog. Clicking on the frame will cause a beep and that's all.

A few months ago, I asked myself the question: Is there a way in which we can find a frame that has already been created?

One of my first newsletters showed a GlobalHotkeyManager that allowed you to install your own event queue into the AWT, letting you catch any events that occur, before ANY other components get hold of them. Why don't we use the concept to catch Window events and then use those events to remember which frames are available?

The problem with the GlobalHotkeyManager, was that it only allowed exactly one event catcher to be present at a time. It is therefore only safe to use if none of the libraries you use link in a similar event queue. An alternative, suggested by F.S., was to register ourselves as a listener to the main AWT event system by calling Toolkit.getDefaultToolkit().addAWTEventListener().

Each time a window is activated, we grap the handle and add it to our list of frames that we know about. We can then write a method called "lookupFrame(String title)" which goes through the list and returns the first frame that we find with the specified title.

```
//: FrameLookup.java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class FrameLookup {
  private final Collection frames = new LinkedList();
  // Singleton Pattern
  private static final FrameLookup instance = new FrameLookup();
  public static FrameLookup getInstance() {
    return instance;
  }
  private FrameLookup() {
    Toolkit.getDefaultToolkit().addAWTEventListener(
      new AWTEventListener() {
        public void eventDispatched(AWTEvent event) {
          System.out.println("Event Dispatched : " + event);
          if (event.getID() == WindowEvent.WINDOW_ACTIVATED) {
            if (event.getSource() instanceof Frame) {
              synchronized(frames) {
                frames.add(event.getSource());
              }
            }
          }
        }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }, AWTEvent.WINDOW_EVENT_MASK);
  }
  public Frame lookupFrame(String title) {
    synchronized(frames) {
      Iterator it = frames.iterator();
      while(it.hasNext()) {
        Frame frame = (Frame)it.next();
        if (frame.getTitle().equals(title)) return frame;
      }
    }
    return null;
  }
}
```

I can now write a test program that creates a frame, forgets the handle, and then creates a dialog and uses the FrameLookup class to find the correct frame to use as owner.

```
//: FrameLookupTest.java
import javax.swing.*;
import java.awt.*;

public class FrameLookupTest {
  private static final String SWING_FRAME_TITLE = "Swing Frame";
  private static final String AWT_FRAME_TITLE = "AWT Frame";
  private static final String SWING_DIALOG_TITLE = "Swing Dialog";

  public static void main(String[] args) {
    FrameLookup framer = FrameLookup.getInstance();
    makeFrame();

    // we can find visible frame by using our FrameLookup utility
    System.out.println("Frame is " +
      FrameLookup.getInstance().lookupFrame(SWING_FRAME_TITLE));

    makeDialog();
  }

  private static void makeFrame() {
    JFrame frame = new JFrame(SWING_FRAME_TITLE);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(300, 300);
    frame.show();
  }

  private static void makeDialog() {
    // we can bind our JDialog to the Frame
    JDialog dialog = new JDialog(
      FrameLookup.getInstance().lookupFrame(SWING_FRAME_TITLE),
      SWING_DIALOG_TITLE, true);
    dialog.setSize(400, 100);
    dialog.show();
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

This is great; we can discover the correct frame as owner, just by knowing its title! There are some minor problems, such as the fact that there is a racing condition involved. It can happen that we request to find the frame before it has been activated or shown, thereby our lookupFrame() method would return "null". Big Oooops. Our situation is now worse than before, as such intermittent failings are MUCH harder to find. Let's also not forget that the title of a frame does not have to be unique.

When I was preparing this newsletter, I got distracted in that I tried to reproduce the Swing dealock problem (which I didn't manage :(. While I was looking through the source code of JFrame/Frame/Window/Container/Component, I happened upon the method Frame[] java.awt.Frame.getFrames(), which Sun added in the JDK 1.2, according to the @version tag. Is it just me, or have I been reading the wrong publications, that don't tell me about these features? Please, if you have heard of Frame.getFrames(), send me an email.

Anyway, with my newly acquired knowledge, I ran off and wrote a second version of the FrameLookup class that looks like this, and sorts out the racing condition problem:

```
//: FrameLookup.java take 2
import java.awt.*;
public class FrameLookup {
  // Singleton Pattern
  private static final FrameLookup instance = new FrameLookup();
  public static FrameLookup getInstance() {
    return instance;
  }
  private FrameLookup() { }
  public Frame lookupFrame(String title) {
    Frame[] frames = Frame.getFrames();
    for (int i=0; i<frames.length; i++) {
      if (frames[i].getTitle().equals(title))
        return frames[i];
    }
    return null;
  }
}
```

We still have the problem that titles alone could be ambigious, but the getFrames() method is definitely the correct way to find a Frame if you need to.

Until next week, and please remember that the more people read this newsletter each week, the more corporate time I can waste collectively, currently I'm wasting about 2 man-months of your development time each week. i.e. please keep on forwarding these newsletters to friends and colleagues who use Java ;-)

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 020] - Serializing Objects Into Database

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 20th issue of "The Java(tm) Specialists' Newsletter", where I look at how we can serialize objects into a database using JDBC. Thank you for your continued support, it makes the hours of research and writing worthwhile :-)

This week's newsletter has been the most frustrating one to write; databases usually have that effect on me. Forgive me if this newsletter seems too elementary, I've actually in the past had to dig into the C code of the JDBC-ODBC bridge to figure out why it wasn't working, i.e. it's more complicated than it seems. Java truly is a "write-once-debug-everywhere" language, especially when it comes to JDBC drivers.

If you're crazy enough to use MSSQL as your database, which isn't THAT crazy considering the beautiful administrative tools available and the ease with which it can be set up, you might consider using the JDBC-ODBC bridge to access the database. It's a dog, full of bugs, but it's free (if you don't count the developer's time in writing workarounds as significant). Also, "better the devil you know", which according to "The Wordsworth Dictionary of Cliche" means "Trust the person or thing you are familiar with rather than risking the unknown." Other JDBC drivers are bound to also contain bugs, and at least we know the bugs in the bridge!

Serializing Objects Into Database

A problem I faced a few months ago, before JDK 1.3 was released with an improved JDBC-ODBC bridge, was how to serialize objects into a relational database using JDBC. At the time, it was quite tricky to do because of bugs in the ODBC bridge. As

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

soon as the object became bigger than 2000 bytes, the method shown below ceased to work. I don't need to mention that I tested it with less than 2000 bytes the first time I wrote the code?

Before we look at how we can write binary objects into a database table, what type of data type should we use? Should we use VARBINARY or IMAGE? The answer, I suppose, depends on how big the object is that you want to write to the database, and which database you're actually talking to (hence my comments regarding write-once-debug-everywhere).

We also have to consider the underlying database when we get a SQLException, so that we can determine what actually went wrong. If the connection goes down temporarily due to a flaky network, it is no good for our application server to crash permanently. In a future newsletter, I will demonstrate how you can find out what went wrong by looking at the SQLException.

Back to the problem of writing serialized Java objects into the database. The most intuitive way of doing it, which as I mentioned before only works in the ODBC bridge of JDK 1.3, is to take the Object, stream it to a ByteArrayOutputStream via an ObjectOutputStream, convert the ByteArrayOutputStream to a byte array and then call the setBytes method on the prepared statement. Note that you have to use PreparedStatement, rather than the normal Statement class, if you want to serialize objects into the database.

To convert it back to a Java object, we simply do the reverse process of reading the byte array, putting it into a ByteArrayInputStream and passing that to an ObjectInputStream. We then read the object, and voila, we have read the object back again.

```
//: DatabaseTest.java
import java.sql.*;
import java.io.*;
import java.util.Vector;
public class DatabaseTest {
  public static void write(
      Object obj, PreparedStatement ps, int parameterIndex)
      throws SQLException, IOException {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    ObjectOutputStream oout = new ObjectOutputStream(baos);
    oout.writeObject(obj);
    oout.close();
    // This will NOT work in JDBC-ODBC bridge under JDK 1.2.2
    // as soon as the size of the byte array is bigger than 2000
    ps.setBytes(parameterIndex, baos.toByteArray());
  }
  public static Object read(ResultSet rs, String column)
      throws SQLException, IOException, ClassNotFoundException {
    // This will NOT work in JDBC-ODBC bridge under JDK 1.2.2
    // as a SQL NULL data value is not handled correctly.
    byte[] buf = rs.getBytes(column);
    if (buf != null) {
      ObjectInputStream objectIn = new ObjectInputStream(
        new ByteArrayInputStream(buf));
      return objectIn.readObject();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
    return null;
  }
  public static void main(String[] args) throws Exception {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection con = DriverManager.getConnection(
      "jdbc:odbc:MailingList", "sa", "");

    Statement st = con.createStatement();
    st.executeUpdate("INSERT BlobTable (Data) VALUES (NULL)");
    st.close();

    PreparedStatement ps = con.prepareStatement(
      "INSERT INTO BlobTable (Data) VALUES (?)");
    write(new Vector(2) {{ add("Hello"); add("World");}}, ps, 1);
    ps.execute();
    Vector veryBig = new Vector(10);
    for (int i=0; i<10; i++) veryBig.add(new byte[10000]);
    write(veryBig, ps, 1);
    ps.execute();
    write("What Gives?", ps, 1);
    ps.execute();
    write(null, ps, 1);
    ps.execute();
    ps.close();

    st = con.createStatement();
    ResultSet rs = st.executeQuery("SELECT Data FROM BlobTable");
    while(rs.next()) {
      System.out.println(read(rs, "Data"));
    }
    rs.close();
    st.close();
  }
}
```

To test this code, you'll have to set up a DSN to point to your MSSQL database (or a real database), and set up a table in your database called BlobTable with an IMAGE type for the column called "Data". In my example, I am writing a SQL NULL value into the table, then a small Vector (using dynamic aggregate initialization described in Newsletter 002) and after that a very big Vector, followed by a String and a Java null object. Lastly, I close the PreparedStatement, and read all the objects back and print them to the screen.

The output on my system is:

```
null
[Hello, World]
[[B@2f0db, [B@12d342, [B@6b97fd, [B@478e57, [B@5224ee, [B@76a746,
[B@5ff48b, [B@2ffc70, [B@663e3d, [B@4901] What Gives?
null
```

All very simple, except that in a real environment we don't really want to use IMAGE types in our database, unless absolutely necessary. We would prefer using

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

VARBINARY as they take less space and form part of the block of data in the row. How do we know the size of a Java object in memory? I've worked out some formulae for "guessing" the size of a Java object by looking at its data members, which is a topic for another newsletter. In the meantime, the easiest is to take the biggest set of data that must be supported by your object and simply try out how much space it will take. There is no "sizeof()" method in Java to do that for you, unfortunately.

Some other issues related to the above problem is that a PreparedStatement is bound to a Connection, which means that we cannot rebind it to another Connection. The result is that we cannot use Connection pooling very well for this, as we then have to construct a new PreparedStatement each time we want to insert a Java object into the database, which defeats the reason for having a PreparedStatement. In the JDBC 3.0 specification, there is support for rebinding PreparedStatements, but we'll have to wait a bit before we'll have JDBC 3.0 compliant drivers.

How do you do this with JDK 1.2.x? You have to use the PreparedStatement.setBinaryStream(...) method to write the binary stream directly into the database. The problem is that when you read the binary stream again, and the database contained a SQL NULL, you get some nasty exceptions that are hard to figure out. Obviously, the JDK 1.3.x ODBC bridge throws different exceptions, so I don't advocate using the JDK 1.2.x bridge for writing objects into databases.

Warning Advanced:
A memory leak, discovered by the best and most ruthless Java bug finder I know, occurs when you call the ResultSet.getTimestamp() method. Rather call ResultSet.getString() and convert the resultant String to a Date object. He's recently been granted permanent residence in our beautiful country of South Africa, which is quite a feat, considering the tough immigration laws. Congratulations!

Please send me your comments / experiences on this. I always appreciate any feedback, both positive and negative, and please remember to send this newsletter to others who might be interested in Java.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 021] - Non-virtual Methods in Java

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 21st issue of "The Java(tm) Specialists' Newsletter". In South Africa the age of adulthood is 21, so I hereby declare this newsletter to be "grown up". No more childish jokes, running in the streets or teasing Sun for writing such a stupid language ;-)

This newsletter serves to illustrate why you should always recompile ALL your code when you get a new build of someone's library. It also, as a side issue, demonstrates how you can write non-virtual methods in Java.

Before I get started, a word of thanks to Dr. Jung for giving me this idea on Monday. I persuaded him to promise me another of his excellent newsletters, which should have been due at least this week, but he managed to deflect my attention with an idea that led to this newsletter.

Please take a few minutes to think of who you know that would be interested in receiving this newsletter and forward it to them.

Non-virtual Methods in Java

In C++ you can mark a method to be virtual", which tells the compiler that you will want to use the most derived method in the object hierarchy. Virtual therefore means that if you have a class A with method f() and a subclass B with the method f(), and you call the method f() on a handle of A pointing to a B, then B's f() gets called. If you left out the "virtual" keyword, it would cause A's f() to get called, i.e. it is bound at compile time, rather than runtime.

In Java, on the other hand, ALL methods are virtual, i.e. the most derived method is always called, unless of course (read on). During many of the Java courses I presented, I was faced with the question from hardened criminals (oh no, I meant C++ programmers) of why Java does not support non-virtual methods (these questions usually get asked by the same guys who ask why Java doesn't support multiple implementation inheritance and operator overloading *groan*).

Let's look at some code:

```
//: A.java
public class A {
  public void f() { System.out.println("A's f()"); }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
//: B.java
public class B extends A {
  public void f() { System.out.println("B's f()"); }
}

//: C.java
public class C {
  public static void main(String[] args) {
    A a = new A();
    a.f();
    ((A)new B()).f();
    B b = new B();
    b.f();
  }
}
```

When we run this, we get the result of:

```
A's f()
B's f()
B's f()
```

A typical question for the Sun Certified Java Programmer exam, and quite obvious to most of us.

The question is, if I have an object of instance B, is it possible to call its parent's f()? Consider class D:

```
//: D.java
import java.lang.reflect.*;
public class D {
  public static void main(String[] args) throws Exception {
    Method f = A.class.getDeclaredMethod("f", new Class[0]);
    f.invoke(new A(), new Object[0]);
    f.invoke((A)new B(), new Object[0]);
    f.invoke(new B(), new Object[0]);
  }
}
```

We are calling f() of class A, but still, the output remains:

```
A's f()
B's f()
B's f()
```

A few months ago, I was asked about this and after battling for a while, gave up and declared that it is not possible to run class C and get the output of:

```
A's f()
A's f()
B's f()
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Now for a little trick that changes all that, we simply make f() private in A and recompile ONLY A:

```
//: A.java
public class A {
  private void f() { System.out.println("A's f()"); }
}
```

We have not recompiled B, C or D, so when we run C, we expect to get some warning or error, but alas, our output for C becomes:

```
A's f()
A's f()
B's f()
```

I tried this out using the JDK 1.3.0_01 and JBuilder 3.0 JDK 1.2 and it worked without problems. Please send me a note if you find a version of Java that somehow gives back a VerifyError or an AccessException.

Our class D does not work, but gives us a runtime error, because A.f() is now private. This is the first time where I've seen reflection resulting in safer code!

When should you use this idea? Please don't, but please at the same time be aware that if you get a new library, even if it's the same version number, you have to recompile every line of your Java code, just to be sure.

A better way of achieving the same goal of non-virtual methods is to use static methods which we then pass a handle to an instance of the class. This much clearer approach was suggested by a founding member of The Contractor's Guild.

```
//: A.java
public class A {
  public static void f(A a) { System.out.println("A's f()"); }
}

//: B.java
public class B extends A {
  public static void f(B b) { System.out.println("B's f()"); }
}

//: C.java
public class C {
  public static void main(String[] args) {
    A a = new A();
    a.f(a);
    a = new B();
    a.f(a);
    B b = new B();
    b.f(b);
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

In summary, always remember to recompile all your classes, since you don't know which ones will be affected by someone else's change!

Warning Advanced:
In JDK 1.1.x, final methods were inlined at compile-time, further necessitating a complete compile if a library was changed.

Thanks for taking the time to read this newsletter, please also take the time to give me feedback, I always appreciate it.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 022] - Classloaders Revisited: "Hotdeploy"

Author: Dr. Christoph G. Jung

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 22nd issue of "The Java(tm) Specialists' Newsletter", sponsored by infor business solutions AG, an ERP company based in Germany. Many thanks go to Christoph Jung for again making an excellent contribution to "The Java(tm) Specialists' Newsletter" by taking the time to write about some of the advanced things he's busy with. This is the longest newsletter so far, and also the most in-depth. Don't start expecting the same quality when I write ;-), in fact, I might skip next week's letter to give you some time to digest this one.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

As always, I would like to ask you to forward this newsletter to anybody you know who might be interested in Java. Please also remember that if you are a beginner in Java, some of these news- letters will go over your head. Don't give up!

The minimalist application server Or Classloading revisited

A few weeks ago, I was in the middle of an ICQ chat with our admired Dr. Kabutz when he suddenly uttered a definite "hi, hi" which took me by surprise as it was different to the ubiquitous "emoticons" or smileys that I am used to. Nevertheless, I was immediately able to catch the thousands of denotations that this very special textual laughter was transporting. In this case, the semantics can be very well subsumed by the German term "verschmitzt" which is difficult to translate to English but would probably be a combination of "roguish", "twisty", "experienced" and "savvy". Similarly, "ha, ha" (a very open reaction to a joke) and "ho, ho" (kind of patronizing Santa Claus stance) are able to express much more context than any smiley construction I know of, at a minimum overhead of bandwidth. Measurements using WinZip have shown verbose laughter statistically being only 5% more expensive than its certainly poorer smiley counterpart.

And now ... I hope you kept a copy of newsletter 18 where Heinz illuminated us with the insight that class identity in the virtual machine is quite independent of even fully-qualified class names. The magical constructs that he used for his investigations were the java.lang.ClassLoader and suitable derivatives, such as the java.net.URLCLassloader. These are, amongst other innovations of the Java(tm) 2 Runtime, indeed a fascinating subject for experimentation.

Classes and ClassLoaders

Perhaps you found that having several classes called "A" in your java.exe was not very useful. On the contrary, it probably required quite an effort to keep the various sources and class files of that newsletter in separate directories. However, I would like to demonstrate how this rather theoretical possibility has significant practical value when it comes to building real programs that are able to simultaneously host multiple Java applications. The most well known examples of such programs are found in the J2EE(tm) application servers.

For this purpose, let us briefly revisit what the head of Maximum Solutions Ltd., by the way, one of the best Java consultancies I know of ;-) [hk: hihi], has demonstrated to our surprised eyes and what may be already apparent from the terminology: A java.lang.ClassLoader is an object that is used to locate external resources (such as *.class file placed in a directory or zipped into a jar-file) and to import the contained byte code for constructing java.lang.Class instances at runtime.

Each java.lang.Class in the JVM is hence associated with its so-called "defining classloader" that has imported it and that can be obtained using the Class.getClassLoader() method. For example, regard the following classes stock.Warehouse (interface to a mono-product warehouse exposing a reservation method) and sales.Order (remotely connects/links to a warehouse to satisfy the order in quantities of size "bunch").

```
// stock/Warehouse.java
package stock;
/** Sample remote interface that is linked by sales.Order. */
public interface Warehouse extends java.rmi.Remote {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  /** Reserve items in this warehouse */
  void reserve(int quantity) throws java.rmi.RemoteException;
}

// sales/Order.java
package sales;
import java.rmi.*;
import stock.Warehouse;
/** Class that links in another jar */
public class Order {
  /** the stock.Warehouse which hosts our items */
  private final Warehouse wHouse;
  /** how much do we need? */
  private final int amount;
  /** quantize reservations */
  private static int bunch=5;
  /** constructs a new order */
  public Order(String itemName, int quantity)
      throws java.net.MalformedURLException, NotBoundException,
        RemoteException {
    // look into the rmi registry to locate the warehouse
    wHouse=(Warehouse)Naming.lookup("//localhost/"+itemName);
    this.amount=quantity;
  }
  /** method that delegates reservation to warehouse */
  public void reserve() throws RemoteException {
    for(int count = 0; count < amount; count += bunch)
      wHouse.reserve(bunch);
  }
}
```

Although sophisticated names and nifty logic should suggest an increasing level of practicability compared to Heinz's previous excursions, please remember that this is a technical newsletter and not a business logic course. Hence, instead of flaming about the obvious flaws and, at the same time, selling the attached files as ready-made ERP-competitor to our infor:COM, compile the sources into two separate directories "classes/stock" and "classes/sales":

```
javac -d classes\stock\ stock\*.java
javac -classpath classes\stock\ -d classes\sales\ sales\*.java
// server/Loader.java
package server;
import java.net.*;
import java.io.*;
public class Loader {
  /** Demonstration of some classloading issues */
  public static void main(String[] args) throws Exception {
    // construct a tiny classloader hierarchy
    ClassLoader stockLoader = new URLClassLoader(
      new URL[] {getStockLocation()});
    ClassLoader salesLoader = new URLClassLoader(
      new URL[] {getSalesLocation()}, stockLoader);
    // load order class
    Class orderC = salesLoader.loadClass("sales.Order");
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    System.out.println(orderC + " loaded from " + salesLoader +
      "; defined by " + orderC.getClassLoader());
    // load warehouse class
    Class wHouseC = salesLoader.loadClass("stock.Warehouse");
    System.out.println(wHouseC + " loaded from " + salesLoader +
      "; defined by " + wHouseC.getClassLoader());
    // analyse class links
    System.out.println("loading and linking same " +
      wHouseC.equals(orderC.getDeclaredField("wHouse").getType()));
    System.exit(0);
  }
  /** where the stock classes can be found */
  protected static URL getStockLocation() throws IOException {
    return new File("classes/stock/").toURL();
  }
  /** where the sales classes can be found */
  protected static URL getSalesLocation() throws IOException {
    return new File("classes/sales/").toURL();
  }
}
```

Whenever the virtual machine (VM) gets hold of your just produced sales.Order - maybe using the loadClass("sales.Order") statement in the preceding server.Loader code - it will also automatically ask the defining classloader of sales.Order to additionally load the linked class into memory:

salesLoader.loadClass("stock.Warehouse").

Since we have placed the byte code of stock.Warehouse into the classes\stock directory that is not under the hood of salesLoader, salesLoader will delegate the call to its parent stockLoader and, upon successful resolution, the respective class-representations in the VM are linked together. These issues will become apparent if you run Loader from the "classes\server" folder (using the attached "lazy developer" policy file which you should never use in production environments, you have been warned!).

```
javac -d classes\server\ server\*.java
java -Djava.security.policy=server.policy -classpath
  classes\server server.Loader
// server.policy
grant {
  // Allow everything for now [hk: *ouch*]
  permission java.security.AllPermission;
};
```

The output will most likely look similarly to this:

```
class sales.Order loaded from java.net.URLClassLoader@253498;
   defined by java.net.URLClassLoader@253498
interface stock.Warehouse defined by java.net.URLClassLoader@209f4e
loading versus linking: true
```

Note that java.exe operates a bit lazily since salesLoader.loadClass("stock.Warehouse") and hence

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

stockLoader.loadClass("stock.Warehouse") will not happen until you begin to inspect or instantiate the imported Order class! In our Loader code, this is implicitly triggered by the getDeclaredField("wHouse") call. This way, you will not end up loading every class file that is under control by any classloader into memory when you were just asking for Order. You will really appreciate this feature if you have experienced the class inflation phenomenon typical in an OO project ;-)

Hot-Deploy

As an educated audience [hk: i.e. you read all my newsletter ;-}] you did know all this before of course. This loading & linking is certainly not restricted to our Order and Warehouse, but is used to intern every class into the VM. It even does this with the main class and the Java(tm) 2 runtime representations that the main class depends upon. The big difference to our example is, however, that you usually specify a single classpath/ classloader ("classes\server\") and a single main class ("server.Loader") at startup of java.exe, hence a single java "application".

With our freshly acquired knowledge, we have revealed the opportunity to dynamically start/shutdown applications at runtime which is often called "deploying": java.exe starts with a minimal setup, e.g., the below server.ApplicationServer, and is then equipped with additional application specifications on-the-fly. An application specification that is processed by the server's deploy(...) and start(...) methods consists of the path to the byte-code, the name of the main class, and the set of initial arguments.

```
// server/ApplicationServer.java
package server;
import java.net.*;
import java.util.*;
import java.io.*;
/** A tiny "application server" */
public class ApplicationServer extends Loader {
 /** this map stores application urls->active-threads */
 final protected Map applications = new HashMap();
 /** the deploy method interns an application
  * @param url jar where the application is packed
  * @param main fully-qualified name of Main class
  * @param args arguments to main method
  * @param parent parent classloader
  * @throws ClassNotFoundException if application not found
  * @return the classloader that has been generated for that app
  */
 public synchronized ClassLoader deploy(final URL url,
     String main, final String[] args, ClassLoader parent)
     throws ClassNotFoundException {

   System.out.println("Deploying "+url);

   // is this a redeploy?
   if(applications.containsKey(url)) // yes: tear it down first
     undeploy(url);

   // generate a new classloader
   final ClassLoader loader = constructClassLoader(
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      new URL[] {url}, parent);
    // load the mainclass
    final Class mainClass = loader.loadClass(main);

    // construct a new "main" thread
    Thread newThread = new Thread() {
     public void run() {
       try{
        // run the main method with the given args
        Class[] params = {String[].class}; // args types
        mainClass.getMethod("main", params).invoke(
          mainClass, new Object[]{args});
        // keep application alive until teared down
        synchronized(this) { this.wait(); }
       } catch(java.lang.reflect.InvocationTargetException e) {
        e.getTargetException().printStackTrace();
       } catch(Exception e) {
       } finally {
        // we lock the appServer
        synchronized(ApplicationServer.this) {
          try{
            // in case that any application error occurs
            // (or the application is to be undeployed)
            System.out.println("Stopping " + url);
            // remove entry if still there
            applications.values().remove(this);
            // call cleanup method
            mainClass.getMethod("stop", new Class[0]).invoke(
              mainClass, new Object[0]);
          } catch(Exception _e) {}
        } // synchronized(appServer.this)
       } // finally
     }}; // method run(); class Thread()

     // set the thread context
     newThread.setContextClassLoader(loader);
     // register application x thread
     applications.put(url, newThread);
     // return classloader
     return loader;
  } // method deploy()

  /** starts an application that has already been deployed */
  public synchronized void start(URL url) {
    System.out.println("Starting " + url);
    ((Thread)applications.get(url)).start();
  }
  /** Undeploys a running application. Never, I repeat, NEVER, do
   *  this using Thread.stop() but use the various options that
   *  are proposed by your JDK documentation to gracefully notify
   *  a thread of shutdown.
   * @param url url where the application is packed
   * @throws Exception if the app could not be teared down
   */
  public synchronized void undeploy(URL url) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    // uh uh. bastard. But for Heinz newsletter, its ok ;-)
    // [hk: gee thanks]
    ((Thread) applications.get(url)).stop(new Exception("stop"));
  }
  /** class loader factory method */
  protected ClassLoader constructClassLoader(URL[] urls,
      ClassLoader parent) {
    return new URLClassLoader(urls,parent);
  }

  /** example usage of the appServer */
  public static void main(String[] args) throws Exception {
    BufferedReader stdin = new BufferedReader(
      new InputStreamReader(System.in));
    ApplicationServer appServer = new ApplicationServer();
    ClassLoader stockLoader = appServer.deploy(
      getStockLocation(), "stock.Main",
      new String[] {"screwdriver", "stock.WarehouseImpl",
        "screwdriver", "200"},
      null);
    appServer.start(getStockLocation());
    stdin.readLine();
    appServer.deploy(getSalesLocation(),
      "sales.Main",
      new String[] {"screwdriver","50"},
      stockLoader);
    appServer.start(getSalesLocation());
    stdin.readLine();
    appServer.deploy(appServer.getSalesLocation(),
      "sales.Main",
      new String[] {"screwdriver","80"},
      stockLoader);
    appServer.start(appServer.getSalesLocation());
    stdin.readLine();
    appServer.undeploy(appServer.getSalesLocation());
    appServer.undeploy(appServer.getStockLocation());
    System.exit(0);
  }
}
```

As example applications that are to be deployed by the presented server logic, we sketch below an exemplary batch sales. Main that constructs and processes a set of sales.Order. And we introduce an exemplary stock application stock. Main that exports remote warehouse services such as implemented by stock.WarehouseImpl.

```
// sales/Main.java
package sales;
/** An example batch application */
public class Main  {
  /** starts the batch */
  public static void main(String[] args) throws Exception {
    // analyse command-line
    for(int count=0; count<args.length; count++)
      // construct order and reserve
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      new Order(args[count++],
        new Integer(args[count]).intValue()).reserve();
    }
}

// stock/WarehouseImpl.java
package stock;
import java.rmi.server.UnicastRemoteObject;
/** Example implementation of a remote service */
public class WarehouseImpl extends UnicastRemoteObject
    implements Warehouse {
  /** number of stored items */
  protected int quantity;
  /** constructs warehouse */
  public WarehouseImpl(String itemName, int quantity)
      throws java.rmi.RemoteException {
    this.quantity=quantity;
  }
  /** reserves items*/
  public void reserve(int amount) {
      System.out.println(this + " is about to reserve " +
        amount + " items.");
      if(quantity < amount) empty(amount - quantity);
      quantity -= amount;
  }
  /** what to do if the warehouse is empty */
  protected void empty(int underLoad)  {
      throw new IllegalArgumentException("warehouse empty");
  }
}

// stock/Main.java
package stock;
import java.rmi.*;
import java.util.*;
/**
 * Example service-publishing application
 */
public class Main {
  /** the services provided by this application */
  protected static Collection services = new Vector();

  /** create and export services */
  public static void main(String[] args) throws Exception {
    System.setSecurityManager(new RMISecurityManager());
    for(int count=0; count<args.length; count++) {
      services.add(args[count]);
      // use context classloader to resolve class names
      Naming.rebind("//localhost/"+args[count++],
        (Remote)Thread.currentThread().getContextClassLoader().
          loadClass(args[count++]).getConstructor(
            new Class[] {String.class,int.class}).newInstance(
            new Object[] {args[count++],
              new Integer(args[count])}));
      // [hk: spend some minutes trying to understand the above
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      // 6 lines - if you do manage to understand you should
      // probably be writing C code?]
    }
  }
  /** tearDown services means unPublish */
  public static void stop() {
    Iterator allServices=services.iterator();
    while(allServices.hasNext()) {
      try{
        Naming.unbind("//localhost/"+allServices.next());
      } catch(Exception e) {}
    }
  }
}
```

If you follow the steps below, you should see these two applications interoperate via a dynamically created classloader hierarchy that is quite similar to the previous example (the RMI-codebase property is required to enable dynamic classloading through rmiregistry; it's value does not matter).

```
javac -d classes\stock\ stock\*.java
rmic -d classes\stock\ stock.WarehouseImpl
javac -classpath classes\stock\ -d classes\sales\ sales\*.java
javac -d classes\server\ server\*.java
start rmiregistry
pause // until rmiregistry is up
java -Djava.security.policy=server.policy
  -Djava.rmi.server.codebase=infor -classpath classes\server
  server.ApplicationServer
```

Pressing <Enter> for the first time outputs:

```
Deploying file:/J:/misc/deployer/classes/stock/
Starting file:/J:/misc/deployer/classes/stock/

Deploying file:/J:/misc/deployer/classes/sales/
Starting file:/J:/misc/deployer/classes/sales/

stock.WarehouseImpl[RemoteStub [ref: [endpoint:[192.168.202.184:2053]
(local),objID:[0]]]] is about to reserve 5 items.
stock.WarehouseImpl[RemoteStub [ref: [endpoint:[192.168.202.184:2053]
(local),objID:[0]]]] is about to reserve 5 items.
stock.WarehouseImpl[RemoteStub [ref: [endpoint:[192.168.202.184:2053]
(local),objID:[0]]]] is about to reserve 5 items.
stock.WarehouseImpl[RemoteStub [ref: [endpoint:[192.168.202.184:2053]
(local),objID:[0]]]] is about to reserve 5 items.
stock.WarehouseImpl[RemoteStub [ref: [endpoint:[192.168.202.184:2053]
(local),objID:[0]]]] is about to reserve 5 items.
stock.WarehouseImpl[RemoteStub [ref: [endpoint:[192.168.202.184:2053]
(local),objID:[0]]]] is about to reserve 5 items.
... etc.
```

We can now experience how flexible our application server has become. After the first sales batch has been executed, please set the static "bunch" variable in

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

sales.Order to "10". Recompile the class and press <Enter> for the second time just to see the changed byte-code running in place of the outdated sales representations. That we could realise this behaviour without having to cycle java.exe, even without affecting the referred stock logic in memory, is called "hot-redeploy" - a very useful feature when it comes to incrementally debug server-side logic or to update customer sites on the fly.

```
Deploying file:/J:/misc/deployer/classes/sales/
Stopping file:/J:/misc/deployer/classes/sales/
Starting file:/J:/misc/deployer/classes/sales/

stock.WarehouseImpl[RemoteStub [ref:
[endpoint:[192.168.202.184:2053](local),objID:[0]]]] is
about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2053](local),objID:[0]]]]
is about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2053](local),objID:[0]]]]
is about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2053](local),objID:[0]]]]
is about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2053](local),objID:[0]]]]
is about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2053](local),objID:[0]]]]
is about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2053](local),objID:[0]]]]
is about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2053](local),objID:[0]]]]
is about to reserve 10 items.
```

Warning: The deprecated Thread.stop() method used in the unDeploy() implementation has allowed me to write concise "newsletter code". Never, I repeat, NEVER use Thread.stop() because it is inherently unsafe: In our example, there is indeed the small opportunity that handling of an application error (an InvocationTargetException reported through reflecting the main method) has not yet obtained the lock to the ApplicationServer.this instance and a simultaneous unDeploy(...) will implant a shutdown exception on top of the cleanup logic. Please, consult your JDK documentation for safe ways to shutdown your threads! [hk: this actually happened once when I ran the tests!]

A second side note: With JDK/JRE 1.3, the apparent relationship between threads and classloaders has been made explicit by associating any thread with a so-called "context classloader" (Thread.setContextClassLoader(ClassLoader loader); Thread.getContextClassLoader()). They are however still unrelated according to 99% of the JDK code which still uses Class.forName(String name) - an awful static method that delegates to the "calling class" defining classloader. The correct way of loading classes in the light of our new knowledge should be always (see ApplicationServer.deploy() and stock.Main.main()):

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Thread.currentThread().getContextClassLoader().loadClass(name)

Alternative Classloading Delegation

There is an unwritten law for middleware engineers that no matter how sophisticated your framework (and with exception of the Thread.stop(), we can be really proud of our ApplicationServer, couldn't we?), after two month of releasing the system basis, the application developers will cooperatively abuse it in ways that you have never imagined before.

Be sure that this st**pid stock team will have (as pressed by the product managers and ... sic! ... customers with ... sicut! ... wishes) compiled another, undoubtfully useful, but technically devastating warehouse implementation that is able to delegate unsatisfied reservation calls to other (vendor) warehouses:

```
// DelegatingWarehouseImpl.java
package stock;
import sales.Order;
/** Example service that links against sales.Order and introduces
 *  a mutual dependency */
public class DelegatingWarehouseImpl extends WarehouseImpl {
  /** name of the item at our vendor */
  protected String vendorItem;
  /** construct a new delegating warehouse */
  public DelegatingWarehouseImpl(String vendorItem, int quantity)
      throws java.rmi.RemoteException {
    super(vendorItem, quantity);
    this.vendorItem = vendorItem;
  }
  /** Overrides the underCapacity reaction to order at vendor */
  protected void empty(int underLoad) {
    try {
      new Order(vendorItem, underLoad).reserve();
      quantity += underLoad;
    } catch(Exception ex) {
      throw new IllegalArgumentException(
        "Could not place order " + ex);
    }
  }
}
  javac -classpath classes\sales\;classes\stock\
    -d classes\stock\ stock\DelegatingWarehouseImpl.java
```

Guess what ... they will phone you and tell you that there is an ugly NoClassDefFoundError: sales.Order thrown when trying to deploy and run the freshly compiled stock (with a voice indicating that this could not be much of a problem for these Java cracks in the tech department, knowhaddimean, nudge, nudge?).

Boom! Your whole nice server-side architecture collapses as SUN does not allow you to have the salesLoader (hosting DelegatingWarehouseImpl and the Warehouse that is needed by Order) both as parent AND child of the stockLoader (hosting Order that is needed by DelegatingWarehouseImpl). They would not call them children and parent, otherwise, would they?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Your options are now (leaving the Java(tm) platform is not an option, stupid! [hk: why actually not?!?]):

Yelling: "This is not the right project structure! We must extract a more abstract masterdata. Order that is deployed in a separate application and that is used by both sales and stock." But, with a tree-like application structure that follows the SUN classloader hierarchy, this cannot be done ad infinitum. You will most likely end up having most classes in masterdata and under the responsibility of a single person (which in any middle-sized company, sits on the same floor as the system basis programmer, hence this would be not a good idea).

Crying: "Forget classloader delegation! We deploy everything into a single URLClassloader by exposing the protected addURL(URL url) method." But then, you loose the hot-deploy feature, again! In order to exchange a tiny class in a tiny module in the server, you will have to tear down and restart the whole logic. Now try to sell your customer these large extra coffee breaks at maintenance time!

Thinking: You sit down silently and release a new server. SmartApplicationServer which implements classloader delegation a bit differently to SUN without your colleagues even noticing except that they will no more detect any NoClassDefFoundErrors when implementing the next of their great ideas while you are having that large extra coffee break, a Cohiba Siglo V [hk: I'd opt for a Partagas Series D] and the new Linux Magazine:

```
// SmartApplicationServer.java
package server;
import java.io.*;
import java.net.*;
import java.util.*;

/** An "application server" that copes with mutual application
 *  dependencies */
public class SmartApplicationServer extends ApplicationServer {
  /** A classloader extension that is able to delegate to other
   *  application~s classloaders */
  protected class SmartClassLoader extends URLClassLoader {
    /** mirrors parent constructor */
    public SmartClassLoader(URL[] urls, ClassLoader parent) {
      super(urls,parent);
    }

    /** dispatch "normal" loadClass method to another name */
    protected Class loadClassNormal(String name, boolean resolve)
        throws ClassNotFoundException {
      return super.loadClass(name,resolve);
    }

    /** override "normal" loadClass method in order to delegate */
    protected Class loadClass(String name, boolean resolve)
        throws ClassNotFoundException {
      try{
        // first we try it traditionally
        return loadClassNormal(name,resolve);
      } catch(ClassNotFoundException e) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      // if this doesnt help, we ask the other application
      // threads for help
      synchronized(SmartApplicationServer.this) {
        Iterator allThreads = applications.values().iterator();
        while(allThreads.hasNext()) {
          SmartClassLoader nextLoader = (SmartClassLoader)
            ((Thread)allThreads.next()).getContextClassLoader();
          if(nextLoader!=null && !nextLoader.equals(this)) {
            try{
              // try the context class loader of next thread
              return nextLoader.loadClassNormal(name,resolve);
            } catch(ClassNotFoundException _e) {
            }
          }
        }
        // they could not help us, hence we throw an exception
        throw new ClassNotFoundException(
          "class could not be found amongst applications.");
      } // synchronized(SmartApplicationServer.this)
    }  // catch
  } // method loadClass()
} // class

/** override parents factory method to construct dedicated
 *  classloaders */
public ClassLoader constructClassLoader(URL[] urls,
    ClassLoader parent) {
  return new SmartClassLoader(urls,parent);
}

/** example of the improved application server */
public static void main(String[] args) throws Exception {
  BufferedReader stdin = new BufferedReader(
    new InputStreamReader(System.in));
  ApplicationServer appServer=new SmartApplicationServer();
  appServer.deploy(getStockLocation(), "stock.Main",
    new String[] {"screwdriver", "stock.DelegatingWarehouseImpl",
    "vendorScrewer", "20", "vendorScrewer", "stock.WarehouseImpl",
    "vendorScrewer","200"},null);
  appServer.deploy(getSalesLocation(), "sales.Main",
    new String[] {"screwdriver","50"}, null);
  appServer.start(getStockLocation());
  stdin.readLine();
  appServer.start(getSalesLocation());
  stdin.readLine();
  appServer.undeploy(getSalesLocation());
  appServer.undeploy(getStockLocation());
  System.exit(0);
 }
}
javac -d classes\server\ server\*.java
java -Djava.security.policy=server.policy
  -Djava.rmi.server.codebase=infor -classpath classes\server
  server.SmartApplicationServer
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Let me elaborate a bit on the steps that we need to take to realise this third, comfortably sounding scenario. The key ingredient is the introduction of a SmartApplication.SmartClassLoader inner class derived from java.net.URLClassLoader that is intimately coupled to the SmartApplicationServer, more specifically, to its "applications" map that stores the running deployments.

In java.lang.ClassLoader, loadClass(String name) is fixedly and simply implemented to call loadClass(name,false) whose task is to not only to locate and intern the class, but also to immediately resolve linked classes if the second boolean parameter is set to true. We can now easily implement a different delegation semantics by overriding the second method:

First we try to call the "standard" classloading semantics which we have re-exposed as loadClassNormal(String name, boolean resolve). If this succeeds we return the found class (at the minimum "overhead" of a try/catch block).

Else (in the above example in which we have removed any classloading hierarchy, this happens both if sales.Order tries to link stock.Warehouse as well as if stock.WarehouseImpl tries to link sales.Order) we catch the ClassNotFoundException and iterate through all other deployed applications to also try loadClassNormal(...) through their main thread's context classloader until the right resource has been found.

Only if the class could not be found in all deployed applications, we throw the final ClassNotFoundException.

By these "minimally invasive" extensions, we finally produce the desired output:

```
Deploying file:/J:/misc/deployer/classes/stock/
Deploying file:/J:/misc/deployer/classes/sales/
Starting file:/J:/misc/deployer/classes/stock/

Starting file:/J:/misc/deployer/classes/sales/

stock.DelegatingWarehouseImpl[RemoteStub [ref:
[endpoint:[192.168.202.184:2269](local),objID:[0]]]]
is about to reserve 10 items.
stock.DelegatingWarehouseImpl[RemoteStub [ref:
[endpoint:[192.168.202.184:2269](local),objID:[0]]]] is about
to reserve 10 items.
 stock.DelegatingWarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2269](local),objID:[0]]]] is
about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub [ref:
[endpoint:[192.168.202.184:2269](local),objID:[1]]]] is about
to reserve 10 items.
 stock.DelegatingWarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2269](local),objID:[0]]]] is
about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub [ref:
[endpoint:[192.168.202.184:2269](local),objID:[1]]]] is about
to reserve 10 items.
 stock.DelegatingWarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2269](local),objID:[0]]]]
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

is about to reserve 10 items.
 stock.WarehouseImpl[RemoteStub
[ref: [endpoint:[192.168.202.184:2269](local),objID:[1]]]]
is about to reserve 10 items.

Stopping file:/J:/misc/deployer/classes/sales/
Stopping file:/J:/misc/deployer/classes/stock/

Ideally, this has been just the beginning of a wonderful exploration into the world of "how can I tweak these suboptimal JDK-classes to fit my practical needs". For example, you will most likely ask how the hell an application programmer or even a customer deployer should be able to manage all those dependencies between applications (usually: jar-files, such as stock.jar and sales.jar) once your system has evolved to a particular size. The solution to this is quite straightforward: Jar dependencies are usually encoded in the Manifest.mf class-path entry and that is where your application server can obtain them in order to identify deployment needs and deployment order.

Similarly, the loadClass calls delegated from one application to another (in a full-blown implementation, also calls to getResource(...) must be delegated) introduce runtime dependencies: If you cycle the byte code of the "defining application" you must also cycle the dependent byte-code of the consuming application. I leave this (together with caching of class locations) as an exercise for you.

Alternatively, if you are a bit lazy, nevertheless curious, i.e., your are a system programmer, you can have a look at the latest org.jboss.deployment.scope.* sources of the JBoss Open Source application server (http://www.jboss.org) whose ingenious mastermind, Marc Fleury, has been the originator of the presented classloading semantics. Besides clustering support, the goal for the next major release of Jboss (3.0 - Project Rabbit Hole) is to have a deployer that can manage multiple scopes, i.e., "virtual applications" such as we have today constructed out of stock.* and sales.*

He, he!

[hk: hihi]

Thanks for taking the time to read this newsletter, please also take the time to give us feedback, we always appreciate it.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 023] - Socket Wheel to handle many clients

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 23rd issue of "The Java(tm) Specialists' Newsletter", where I try and get back to my roots of distributed performance evaluation. My PhD thesis was entitled "Analytical Performance Evaluation of Concurrent Communicating Systems using SDL and Stochastic Petri Nets", or something like that. The main idea was to automatically map protocols designed in the Specification and Description Language (SDL) to a modelling language called Stochastic Petri Nets for the simple reason that there are a lot of well known analytical techniques available for evaluating a Stochastic Petri Net's performance. But, all that was of another era when it did not matter that it would take a very long time to analyse a protocol of any real size. In fact, that was last millenium, i.e. long long ago, i.e. don't bother asking me any questions about it ;-)

When we measure performance we have to consider mainly two criteria: memory and cpu cycles. I was able to significantly reduce the amount of memory needed for the server by the idea presented here, but I did not manage to increase the speed at which clients are serviced, although it converged. If you think of anything that would improve the speed of the SocketWheel, please let me know, and you will earn instant fame in over 40 countries by being immortalized in my next newsletter.

Please forward this newsletter to as many people as you know who are interested in programming in Java at more-than-entry-level.

Socket Wheel to handle many clients

The typical way of implementing a server that needs to "talk back" to the client is to construct a thread for each client that is connected, normally through a thread pool. For example, consider the Server.java file:

```
// Server.java
import java.net.*;
import java.io.*;
public class Server {
  public static final int PORT = 4444;
  public Server(int port) throws IOException {
    ServerSocket ss = new ServerSocket(port);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    while(true) {
      new ServerThread(ss.accept());
    }
  }
  private class ServerThread extends Thread {
    private final Socket socket;
    public ServerThread(Socket socket) {
      this.socket = socket;
      start();
    }
    public void run() {
      try {
        ObjectOutputStream out = new ObjectOutputStream(
          socket.getOutputStream());
        ObjectInputStream in = new ObjectInputStream(
          socket.getInputStream());
        while(true) {
          in.readObject();
          out.writeObject(new String("test"));
          out.flush();
          out.reset();
        }
      } catch(Throwable t) {
        System.out.println("Caught " + t + " - closing thread");
      }
    }
  }
  public static void main(String[] args) throws IOException {
    new Server(PORT);
  }
}
```

What this does is simply read an object and write an object for the duration of the client being connected to the thread. When the client disconnects, the thread will stop. The code is not very "clean", I should handle closing of Sockets better than it is, but I don't want to cloud over the issue at stake here.

A client would typically look like this (send an object, read an object, wait some time, etc.):

```
// Client.java
import java.net.*;
import java.io.*;
public class Client {
  public Client(int port) throws Exception {
    Socket socket = new Socket("localhost", port);
    ObjectOutputStream out = new ObjectOutputStream(
      socket.getOutputStream());
    ObjectInputStream in = new ObjectInputStream(
      socket.getInputStream());
    for (int i=0; i<10; i++) {
      out.writeObject(new Integer(i));
      out.flush();
      out.reset();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      System.out.println(in.readObject());
      Thread.sleep(1000);
    }
  }
  public static void main(String[] args) throws Exception {
    new Client(Server.PORT);
  }
}
```

In order to test what happens when a lot of clients connect I wrote a MultiClient class, which constructs 3500 sockets and an equivalent number of object output and input streams to use for sending messages. It then cycles through the sockets and writes to them one object, then cycles through them again and reads one object. The reason I took 3500 sockets is that on my little notebook I could not open more than 3500 sockets, don't ask why, I don't know. That number is *probably* system dependent, so if you get an exception when trying to create a socket, try what happens when you have less sockets.

```
// MultiClient.java
public class MultiClient {
  public MultiClient(int port) throws Exception {
    long time = -System.currentTimeMillis();
    Socket[] sockets = new Socket[3500];
    ObjectOutputStream[] outs =
      new ObjectOutputStream[sockets.length];
    ObjectInputStream[] ins =
      new ObjectInputStream[sockets.length];
    for (int i=0; i<sockets.length; i++) {
      sockets[i] = new Socket("localhost", port);
      outs[i] = new ObjectOutputStream(
        sockets[i].getOutputStream());
      ins[i] = new ObjectInputStream(
        sockets[i].getInputStream());
    }
    System.out.println("Constructed all sockets");
    for (int j=0; j<32; j++) {
      long iterationTime = -System.currentTimeMillis();
      for (int i=0; i<sockets.length; i++) {
        outs[i].writeObject(new Integer(i));
        outs[i].flush();
        outs[i].reset();
      }
      System.out.println(j + ": Written to all sockets");
      for (int i=0; i<sockets.length; i++) {
        ins[i].readObject();
      }
      System.out.println(j + ": Read from all sockets");
      iterationTime += System.currentTimeMillis();
      System.out.println(j + ": Iteration took " +
        iterationTime + "ms");
    }
    time += System.currentTimeMillis();
    System.out.println("Writing to " + sockets.length +
      " sockets 32 times took " + time + "ms");
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }
  public static void main(String[] args) throws Exception {
    new MultiClient(Server.PORT);
  }
}
```

This all works quite nicely, except that each thread in the JDK1.3 implementation of the VM takes up 20KB for its stack. When you add up all the other memory taken up for streams and sockets, it comes to 97MB used up on the server, just to handle a paltry 3500 clients! So, if we had 35000 clients connecting, with the presumption that our machine can handle that many sockets, we not only create 35000 threads (which each don't do that much, but it still takes a long time to construct them all, even if you use a thread pool) but we also gobble up almost 1GB of memory!!!

I was pondering this problem a few days ago and came up with an idea to use polling (yes, I know, polling sucks) to avoid making a thread for each client. Instead of having a server which uses a thread for each client that gets connected, we have a list of sockets that are connected to the server. Writing to a socket will block if the TCP buffer is full, which is about 64000 bytes, so we could have the problem of the server being hung up completely if the client decides to not service the stream. I will conveniently ignore that problem in this newsletter.

We want to have an ObjectOutputStream and an ObjectInputStream associated with each Socket, so we make an inner class to contain those values, which we call a SocketBucket. To make connecting fast, we keep two lists of SocketBuckets, one for the new sockets and one for the already connected sockets. We then run through all the sockets and try to read from each of them with a timeout of 1 millisecond. If there is nothing to read we get an InterruptedIOException and go to the next socket.

If we found at least one socket that had some data, we immediately go looking again, otherwise we go dream for a while and then go looking for more data. There are many different ways in which we could tune this approach, for example, you could keep a set of the last sockets which had data and push their priority up or down, depending on stochastic prediction techniques (not that I remember what that means - just sounded cool!). One of the disadvantages with this approach is also that the server has to wait for an entire millisecond before looking at the next socket. It would be much better to wait less, otherwise if you have 1000 sockets connected, it will take 1 second just to check if any of the sockets have data waiting. Unfortunately, 1 ms is the shortest that we can wait with Java sockets.

Here is the code for the SocketWheel:

```
// SocketWheel.java
import java.net.*;
import java.io.*;
import java.util.*;
public class SocketWheel {
  // the list contains SocketBuckets
  private final List sockets = new LinkedList();
  // we don't want to block a new connection while we are busy
  // serving the existing ones
  private final List newSockets = new LinkedList();
  public SocketWheel() {
    new ServerThread();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }
  public void addSocket(Socket socket) throws IOException {
    synchronized(newSockets) {
      newSockets.add(new SocketBucket(socket));
      newSockets.notify();
    }
  }
  private class SocketBucket {
    public final Socket socket;
    public final ObjectOutputStream out;
    public final ObjectInputStream in;
    public SocketBucket(Socket socket) throws IOException {
      this.socket = socket;
      out = new ObjectOutputStream(socket.getOutputStream());
      in = new ObjectInputStream(socket.getInputStream());
      socket.setSoTimeout(1);  // VERY short timeout
    }
  }
  private class ServerThread extends Thread {
    public ServerThread() {
      super("ServerThread");
      start();
    }
    public void run() {
      long dreamTime = 10;
      boolean foundSomething;
      while(true) {
        try {
          synchronized(newSockets) {
            sockets.addAll(newSockets);
            newSockets.clear();
          }
          foundSomething = false;
          Iterator it = sockets.iterator();
          while(it.hasNext()) {
            SocketBucket bucket = (SocketBucket)it.next();
            try {
              bucket.in.readObject();
              foundSomething = true;
              bucket.out.writeObject(new String("test"));
              bucket.out.flush();
              bucket.out.reset();
            } catch(InterruptedIOException ex) {
              // just skip this socket
            } catch(IOException ex) {
              it.remove();
            }
          }
          if (foundSomething) {
            dreamTime = 6;
          } else {
            if (dreamTime < 1000)
              dreamTime *= 1.5;
            else dreamTime = 1000;
            synchronized(newSockets) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
            // only sleep if we didn't find anything
            newSockets.wait(dreamTime);
          }
        }
      } catch(Throwable t) {
        System.out.println("Caught " + t + " - remove socket");
      }
    }
  }
 }
 public static void main(String[] args) throws IOException {
  SocketWheel wheel = new SocketWheel();
  ServerSocket ss = new ServerSocket(Server.PORT);
  while(true) {
   Socket socket = ss.accept();
   wheel.addSocket(socket);
  }
 }
}
```

When I connect to the SocketWheel server with the MultiClient, the server uses up only 32MB of RAM, basically one third of the other server, but it is a little bit slower to use this approach as opposed to threading, and it is a lot more complicated. In addition, the whole example sometimes gets stuck, I don't know why. If the MultiClient stops proceeding and the CPU goes to 0%, you'll have to restart the MultiClient. (If you spot the problem, please let me know, I suspect it's an underlying C implementation problem which is why I'm not pursuing it.)

With the SocketWheel, the test took 2:36 minutes to complete, the normal threaded Server took only 2:23 minutes to complete, CPU was at 100% both times, disk usage was 0%. The difference in speed is not that great, whereas the memory usage is only 34MB in the SocketWheel server, i.e. roughly 1/3 of the threaded server.

When we change the MultiClient to only use 350 sockets, the SocketWheel takes 14 seconds, the threaded server only 11 seconds, the SocketWheel uses 9.4MB, the threaded server 14MB, a smaller difference, probably because that includes the total memory used by java.exe measured with the task manager, i.e. it includes the memory used by the JVM.

It was quite fun writing this SocketWheel, and was actually a lot easier than I thought possible. Please tell me if you've done something similar or if you think of ways to improve the speed of the SocketWheel server. I tried having a couple of threads in the SocketWheel, but the whole system just got stuck more often and did not improve speed. Please don't use the SocketWheel "as is" unless you're willing to discover and fix the bug that makes it get stuck and cater for clients not reading their sockets.

I always appreciate any feedback, both positive and negative, so please keep sending your ideas and suggestions. Please also remember to take the time to send this newsletter to others who are interested in Java.

Heinz

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 024] - Self-tuning FIFO Queues

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 24th issue of "The Java(tm) Specialists' Newsletter". Winter has arrived in Cape Town with great rain and cold temperatures, much to the chagrin of the author. The only way to combat such is to sit next to the heater with a glass of South African 1997 Zonnebloem Cabernet Sauvignon and write Java newsletters. This newsletter has taken the most time to date - performance tests tend to have that effect on me :(((

Before I go on today's voyage, some feedback regarding the Socket Wheel idea:

I'm not quite sure who gave the best contributions regarding possible improvements for last newsletter's SocketWheel. Neil Bartlett (Algorithmics Canada) and John Vlissides (IBM USA) pointed out that the design was very close to the Reactor pattern by Schmidt. This is a great plus mark for patterns; they truly are things that OO developers around the world would come up with independently. Josh Rehman pointed out that it is not too clever to try minimze memory as that is very cheap nowadays, which I agree with. He also pointed out that JDK 1.4 has non-blocking IO making it possibly a lot easier to implement such a SocketWheel. Ecce Jezuch (Poland) suggested using the available() method to find out how many bytes would be available without blocking, but unfortunately under Windows the JDK always returned 0.

James Pereira provided some excellent information regarding sockets. It's quite technical, so I'm including it verbatim:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

"Registered Ports, ports between 1024 and 49151, are listed by the IANA and on most systems can be used by applications or programs executed by users. Table C.2 specifies the port used by the server process as its contact port. The IANA registers uses of these ports as a convenience to the Internet community. To the extent possible, these same port assignments are used with UDP. The Registered Ports are in the numerical range of 1024-49151. The Registered Ports between 1024 and 5000 are also referred to as the Ephemeral Ports. At least on Windows , The TCP/Stack (OS) re-uses these ports internally on every socket connection cycling from 1024...5000 wrapping around to 1024 again. This could lead to some interesting problems if sockets are opened and close very quickly as there is usually a time delay before that port is made available again...

"Second, the number of user-accessible ephemeral ports that can be used to source outbound connections is configurable with the MaxUserPort registry entry (HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters). By default, when an application requests any socket from the system to use for an outbound call, a port between the values of 1024 and 5000 is supplied. You can use the MaxUserPort registry entry to set the value of the highest port number to be used for outbound connections. For example, setting this value to 10000 would make approximately 9000 user ports available for outbound connections. For more details, see RFC 793. See also the MaxFreeTcbs and MaxHashTableSize registry settings (HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters).

"BTW: The stack and memory size issue is only an NT issue. On NT the stack size is configurabale via linker and EditBin.exe unfortunately we don't want to mess with the VM."

This information should solve the problem that I encountered with too many sockets causing exceptions.

Ideally we should be able to specify stack size per thread as some threads will need a lot and others only a little, so I still think that we have a problem with many threads and their stack size.

Another excellent contribution was made by Craig Larman, author of "Java 2 Performance and Idiom Guide", who suggested using an approach of multiplexing sockets described in his book to minimize the number of sockets needed per client connection. Next week I will try and write about this idea and give you a solution that hopefully will work acceptably. I always recommend Craig's book as an excellent book for the Java fundi who wants to have his mind activated regarding performance ideas. It is very difficult to write anything about performance as it is so dependent on your implementation and hardware. This leads me to this week's newsletters on self-tuning FIFO queues....

Self-tuning FIFO Queues

HotSpot(tm) has caused a lot of trouble for the Java Specialist in the field, especially those of us with a few years experience. All of a sudden, all the hard-earned performance knowledge was wiped out in one foul swoop. The only thing left for us to do was to write the simplest code that could possibly work and let the HotStop compiler sort out performance for us. I liken it to the great financial crisis of Japan in the 90's, where no one knew whether he was coming or going and all the old certainties went out of the window. Luckily, unlike in Japan, we are not taking this

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

too seriously, so our windows are still closed. Fortunately everyone knows that Java is slow ;-)

A lot of the performance tricks we used in old code actually make our code slower under HotSpot. Since we don't know what the performance of our code will be for a specific platform, would it be completely hairbrained to write self-tuning code? Write 3 algorithms, let the program measure on the target platform which performs best, and choose that algorithm for the duration of the VM.

To illustrate this idea, I want to write a FIFO queue that is based on a java.util.List implementation. A while ago I discovered that java.util.ArrayList is sometimes faster than java.util.LinkedList for FIFO queue implementations. The switch over occurs at a specific point in time, which we can measure beforehand. The switch-over point is dependent on the VM, whether we are using HotSpot(tm), etc. For example, on my little notebook with 256MB RAM and Pentium III 700, the cross-over point is 50 elements in the list when I use HotSpot, but 500 elements when I switch off hotspot compiling (sometimes!).

The interface that the FIFO queues will implement is very simple:

```
public interface FIFO {
  /** Add an object to the end of the FIFO queue */
  boolean add(Object o);
  /** Remove an object from the front of the FIFO queue */
  Object remove();
  /** Return the number of elements in the FIFO queue */
  int size();
}
```

We implement this interface and extend ArrayList and LinkedList:

```
// FIFOArrayList.java
import java.util.ArrayList;
public class FIFOArrayList extends ArrayList implements FIFO {
  public Object remove() {
    return remove(0);
  }
}
// FIFOLinkedList.java
import java.util.LinkedList;
public class FIFOLinkedList extends LinkedList implements FIFO {
  public Object remove() {
    return remove(0);
  }
}
```

We also write a SwappingOverFIFOQueue which has values for HIGH and LOW water marks. When we reach a HIGH water mark and we are busy using an ArrayList, we start using a LinkedList. On the contrary, if we reach a LOW water mark and we are busy using a LinkedList we start using an ArrayList.

In foresight to my next example, I have made it possible to set the watermarks, which also checks the optimal list types for all the lists currently in the system. We have to be careful to not get a memory leak by keeping handles to the instances of

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

the SwappingOverFIFOQueue so we use WeakReferences to hold the references. Have a look at newsletter 15 for a discussion on Weak / Soft References.

```
// SwappingOverFIFOQueue.java
import java.util.*;
import java.lang.ref.WeakReference;
public final class SwappingOverFIFOQueue implements FIFO {
  /** The low value after which we switch over to ArrayList */
  private static int LOW = 30;
  /** The high value after which we switch down to LinkedList */
  private static int HIGH = 70;
  /** This list contains weak references of instances of this
      class */
  private static List instances = new LinkedList();
  /** We add the weak references in an initializer block */
  {
    instances.add(new WeakReference(this));
  }
  /** When we set the low and high water marks we go through all
      the existing instances and check for the optimal list type.
      If the references is null we remove the WeakReference from
      our instance list. */
  public static void setWaterMarks(int low, int high) {
    LOW = low;
    HIGH = high;
    Iterator it = instances.iterator();
    while(it.hasNext()) {
      WeakReference ref = (WeakReference)it.next();
      SwappingOverFIFOQueue q = (SwappingOverFIFOQueue)ref.get();
      if (q == null) {
        it.remove();
      } else {
        q.checkOptimalListType();
      }
    }
  }

  private List list = new ArrayList();
  public Class getListType() { return list.getClass(); }
  private int size = 0;
  public boolean add(Object o) {
    try {
      list.add(o);
      return true;
    } finally {
      if (++size == HIGH) checkOptimalListType();
    }
  }
  public Object remove() {
    try {
      return list.remove(0);
    } finally {
      if (--size == LOW) checkOptimalListType();
    }
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public int size() {
    return size;
  }
  private void checkOptimalListType() {
    if (size >= HIGH && (!(list instanceof LinkedList))) {
      list = new LinkedList(list);
    } else if (size <= LOW && (!(list instanceof ArrayList))) {
      list = new ArrayList(list);
    }
  }
}
```

My test program takes the number of entries in the queue and then illustrates how often we can add/remove in 2 seconds for each of the queues. I found that you get the best performance results when you run your tests for about 2 seconds each, so I count iterations rather than milliseconds.

```
// SwappingOverFIFOQueueTest.java
import java.util.*;
public class SwappingOverFIFOQueueTest {
  private static int ENTRIES;
  public static void test(FIFO queue) {
    for (int i=0; i<ENTRIES; i++) {
      queue.add(new Object());
    }
    long up_to = System.currentTimeMillis() + 2000;
    int iterations = 0;
    while(System.currentTimeMillis() <= up_to) {
      queue.add(new Object());
      queue.remove();
      iterations++;
    }
    System.out.println(queue.getClass());
    System.out.println("\t" + iterations + " iterations");
  }
  public static void main(String[] args) {
    if (args.length != 1) {
      System.out.println(
        "Usage: java SwappingOverFIFOQueueTest entries");
      System.exit(1);
    }
    ENTRIES = Integer.parseInt(args[0]);
    System.out.println("Entries = " + ENTRIES);
    test(new FIFOArrayList());
    test(new FIFOLinkedList());
    SwappingOverFIFOQueue q = new SwappingOverFIFOQueue();
    test(q);
    System.out.println("Current queue implementation " +
      q.getListType().getName());
  }
}
```

On my notebook, when I run this program with 0 entries, I get the following output:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
Entries = 0
class FIFOArrayList
        4552883 iterations
class FIFOLinkedList
        2551017 iterations
class SwappingOverFIFOQueue
        3594810 iterations
```

With 100 entries I get:

```
Entries = 100
class FIFOArrayList
        1800877 iterations
class FIFOLinkedList
        2509328 iterations
class SwappingOverFIFOQueue
        2158451 iterations
```

And with 10000 entries I get:

```
Entries = 10000
class FIFOArrayList
        49500 iterations
class FIFOLinkedList
        812933 iterations
class SwappingOverFIFOQueue
        758657 iterations
```

We can thus see that the SwappingFIFOQueue is always faster than the worst case and slower than the best case, as one would logically expect. However, I chose the HIGH and LOW values from some tests that I made on my notebook, for that specific JVM. If I take the JDK 1.2.2 that comes with JBuilder, for 100 entries I get:

```
Entries = 100
class FIFOArrayList
        1434122 iterations
class FIFOLinkedList
        1307108 iterations
class SwappingOverFIFOQueue
        1178115 iterations
```

Or if I use the -Xint mode for JDK 1.3 under Windows to switch off the hotspot compiler, for 100 entries I get

```
Entries = 100
class FIFOArrayList
        497550 iterations
class FIFOLinkedList
        480599 iterations
class SwappingOverFIFOQueue
        392314 iterations
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

In both cases, the values of the SwappingOverFIFOQueue were worse than for both the ArrayList and the LinkedList.

We therefore write a Profiler that finds ideal HIGH/LOW water marks for the JVM that is running on your system and sets up the SwappingOver water marks.

```
// SwappingOverFIFOQueueProfiler.java
import java.util.*;
/*
  For the sake of brevity we only consider two implementations
  of java.util.List, namely java.util.ArrayList and
  java.util.LinkedList. */
public class SwappingOverFIFOQueueProfiler {
  private static boolean isArrayListFaster(int entries) {
    System.out.println("isArrayListFaster(" + entries + ")");
    return countIterations(new ArrayList(), entries) >
      countIterations(new LinkedList(), entries);
  }
  private static int countIterations(List list, int entries) {
    for (int i=0; i<entries; i++) {
      list.add(new Object());
    }
    long end = System.currentTimeMillis() + 1000;
    int iterations = 0;
    while(System.currentTimeMillis() <= end) {
      iterations++;
      list.add(new Object());
      list.remove(0);
    }
    return iterations;
  }

  static {
    int checks = 0;
    int watermark = 1;
    int bestWatermark = 0;
    for (int i=0; i<16; i++) {
      if (isArrayListFaster(watermark)) {
        bestWatermark = Math.max(watermark, bestWatermark);
        watermark *= 2.0;
      } else {
        watermark *= 0.75;
        if (watermark <= bestWatermark)
          watermark *= 1.25;
      }
    }
    System.out.println("Best watermark = " + bestWatermark);
    int low = (int)(bestWatermark * 0.75);
    int high = (int)(bestWatermark * 1.25);
    System.out.println("Setting LOW to " + low +
      " and HIGH to " + high);
    SwappingOverFIFOQueue.setWaterMarks(low, high);
  }
  public static void main(String[] args) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    SwappingOverFIFOQueueTest.main(new String[] { "0" });
    SwappingOverFIFOQueueTest.main(new String[] { "100" });
    SwappingOverFIFOQueueTest.main(new String[] { "10000" });
  }
}
```

If we load this class in our system then it will do measurements of where the swap-over between ArrayList and LinkedList performance occurs. On my computer, with JDK 1.3 and HotSpot, the swap-over was measured to happen at about 32 entries in the list. When I switch off the HotSpot, it occurs at about 121 entries, and under JDK 1.2.2 it happens at about 303.

After spending about 10 hours on this stupid newsletter, I have to conclude that it would be better to stick to a LinkedList for a FIFO queue as it is a better "average" performer. Perhaps the lesson I've learnt from this newsletter is that we must be careful of writing code which is too complicated as it tends to be more difficult to optimize. As performance guru Craig Larman pointed out though, we must be sure to not ignore performance altogether; our customers might just kill the project if the prototypes perform like dogs.

I always appreciate any feedback, both positive and negative, so please keep sending your ideas and suggestions. Please also remember to take the time to send this newsletter to others who are interested in Java.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 025] - Final Newsletter

Author: Dr. Heinz M. Kabutz

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 25th issue of "The Java(tm) Specialists' Newsletter". I hope that for at least *some* of you, your heart sank when you saw the title of this newsletter. No, your mailbox is not getting lighter, I just thought I'd write a bit about how I use the "final" keyword. Incidentally, on Monday we broke through the 1000th reader barrier (on an upward trend) so thanks to all of you who promoted this newsletter and sent it to friends and colleagues. Please remember to forward this newsletter to as many Java enthusiasts as you know who might be interested in receiving such a newsletter.

The last few newsletters where quite heavy, so this week I would like to look at style, specifically on uses of the "final" keyword. It is not a newsletter on how to abuse the "final" keyword, which might surprise some of the more loyal readers of this newsletter.

I would like to thank Carl Smoritcz from Germany for hosting an archive of my newsletters at http://www.smotricz.com/kabutz. Please let me know if you would like to include an archive on your website. I am currently in discussion with a designer to put together a website for my company, which will include the newsletters and information about the type of work my company does.

Final

The keyword "final" in Java is used in different ways depending on the context. We can have final methods, final classes, final data members, final local variables and final parameters. A final class implicitely has all the methods as final, but not necessarily the data members. A final class may not be extended, neither may a final method be overridden.

Final primitive data members cannot be changed once they are assigned, neither may final object handle data members (Vector, String, JFrame, etc.) be reassigned to new instances, but if they are mutable (meaning they've got methods that allow us to change their state), their contents may be changed. Since String is immutable, once a handle to it is final, we *could* consider it as a constant, if we ignore the effects of newsletter 14. So how do we use this construct in the real world?

Final Methods

I personally try to avoid making a method final, unless there is a very good reason for it to be final. There are typically two reasons to make a method final, performance and design. Let's look at performance first:

When a method is final, it may be inlined. Before HotSpot compiling (JDK 1.1.x), these methods were usually inlined at compile time, whereas with HotSpot they are inlined at runtime, unless the compiler can guarantee that the inlined method will always be compiled together with the code that uses it.

// somebody else's class

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
public class A {
  public static final void f() {
    System.out.println("A's f()");
  }
}

// our class
public class B {
  public void g() {
    A.f();
  }
}
```

In the past, at compile time our class would be turned into:

```
// compiled class
public class B {
  public void g() {
    System.out.println("A's f()");
  }
}
```

The effect of this was that we had to make one less method call, and since method calls produce extra overhead, we saved some clock cycles. The disadvantage of this, made clear to me by an old (ok, experienced) COBOL programmer during one of my courses, was that whenever somebody else's class changed we would have to remember to recompile our class!

In JDK 1.[234].x with HotSpot(tm), Sun changed this so that the methods were no longer inlined at compile time, but rather by the HotSpot compiler at run time, IFF the performance measurements suggested that it would improve the overall performance of our code.

There are quite a few factors which will affect whether a method will be inlined or not, and we cannot assume that just because we make something final that it will definitely be inlined. As we saw in newsletter 21, it is a good idea anyway to always recompile all your code when you get a new version of someone else's library, so this is not necessarily a reason to NOT use final methods.

When you make a method final, no-one else will be able to override it again. You thus limit extensibility of your code by choosing to make the method or, even worse, your class final. I have been utterly frustrated in the past when I wanted to extend code where the developer had tried to add optimizations in the form of final. I thus never make a method or class final unless I specifically want to stop do others from overriding them.

So, when do I use final methods? If I have performance values that prove that final makes a difference then I would consider using it for performance reasons, otherwise I would only ever use it for design reasons.

This is all old hat for you I'm sure, so let's look at final data members:

Final data members

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

One of the difficulties in programming is coming up with good names for "things". (there, that just proves my point, doesn't it?) I remember an experienced (ok, old) C programmer who was programming in Java and decided to use very long names for everything, for example:

```
public class SessionConnectorWithRetryAtLeastThreeTimes {
  private String connectionNameReceivedFromInternet;
  private int numberOfTimesThatWeShouldRetryAtLeast;
}
```

Alright, I'm exaggerating a little bit, but I hope you get the idea. The beauty of good names is that comments become very easy to write, sometimes even partly redundant. In Java, we can then write a constructor that takes the state for the object and assigns the correct data members. For example:

```
public class SessionConnectorWithRetryAtLeastThreeTimes {
  private String connectionNameReceivedFromInternet;
  private int numberOfTimesThatWeShouldRetryAtLeast;
  public SessionConnectorWithRetryAtLeastThreeTimes(
String c, int n) {
    connectionNameReceivedFromInternet = c;
    numberOfTimesThatWeShouldRetryAtLeast = n;
  }
}
```

The problem with our constructor is that we have to explain in our documentation what c and n represent. It would be much better to use the same names in the parameters of the constructor as we use for the data members, as it reduces the confusion. The standard way in Java of solving this problem is to use the same names for the parameters as we do for the data members and then to explicitely specify what we are referring to, using the "this" keyword.

```
public class SessionConnectorWithRetryAtLeastThreeTimes {
  private String connectionNameReceivedFromInternet;
  private int numberOfTimesThatWeShouldRetryAtLeast;
  public SessionConnectorWithRetryAtLeastThreeTimes(
String connectionNameReoeivedFromInternet,
int numberOfTimesThatWeShouldRetryAtLeast) {
    this.connectionNameReceivedFromInternet =
connectionNameReceivedFromInternet;
    this.numberOfTimesThatWeShouldRetryAtLeast =
numberOfTimesThatWeShouldRetryAtLeast;
  }
}
```

The above code will compile and run, but not correctly. Take a few minutes to figure out what could possible be wrong with it...

.

.

.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

.

I hope you didn't find the mistake. The first parameter of the constructor is spelt differently to the data member, thanks to a simple spelling mistake. When we thus say

```
  this.connectionNameReceivedFromInternet =
    connectionNameReceivedFromInternet;
```

both the names refer to the data member, so the data member will always null!

This is the reason why some companies try to pursuade their staff to augment their data members with strange characters (m_ or _) to differentiate them from parameters, rather than use the "this" trick. I know of at least two companies where such coding standards are used. The effect is that either the data members look ugly, or the parameters look ugly.

A simple way of preventing such mistakes, besides learning to touch type 100% correctly, is to make the data members final, where possible. When we do that, the code below will no longer compile, and we can find our mistakes much easier.

```
public class SessionConnectorWithRetryAtLeastThreeTimes {
  private final String connectionNameReceivedFromInternet;
  private final int numberOfTimesThatWeShouldRetryAtLeast;
  public SessionConnectorWithRetryAtLeastThreeTimes(
String connectionNameReoeivedFromInternet,
int numberOfTimesThatWeShouldRetryAtLeast) {
    this.connectionNameReceivedFromInternet =
connectionNameReceivedFromInternet;
    this.numberOfTimesThatWeShouldRetryAtLeast =
numberOfTimesThatWeShouldRetryAtLeast;
  }
}
```

As a matter or habit, I make all data members final wherever that is possible. Mistakes that would have taken me days to find now pop out at the next compile.

Final local variables

There are two reasons I know for making a local variable or a parameter final. The first reason is that you don't want your code changing the local variable or parameter. It is considered by many to be bad style to change a parameter inside a method as it makes the code unclear. As a habit, some programmers make all their parameters "final" to prevent themselves from changing them. I don't do that, since I find it makes my method signature a bit ugly.

The second reason comes in when we want to access a local variable or parameter from within an inner class. This is the actual reason, as far as I know, that final local variables and parameters were introduced into the Java language in JDK 1.1.

```
public class Access1 {
  public void f() {
    final int i = 3;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    Runnable runnable = new Runnable() {
public void run() {
  System.out.println(i);
}
    };
  }
}
```

Inside the run() method we can only access i if we make it final in the outer class. To understand the reasoning, we have to look at what the compiler does. It produces two files, Access1.class and Access1$1.class. When we decompile them with JAD, we get:

```
public class Access1 {
  public Access1() {}
  public void f() {
    Access1$1 access1$1 = new Access1$1(this);
  }
}
```

and

```
class Access1$1 implements Runnable {
  Access1$1(Access1 access1) {
    this$0 = access1;
  }
  public void run() {
    System.out.println(3);
  }
  private final Access1 this$0;
}
```

Since the value of i is final, the compiler can "inline" it into the inner class. It perturbed me that the local variables had to be final to be accessed by the inner class until I saw the above.

When the value of the local variable can change for different instances of the inner class, the compiler adds it as a data member of the inner class and lets it be initialised in the constructor. The underlying reason behind this is that Java does not have pointers, the way that C has.

Consider the following class:

```
public class Access2 {
  public void f() {
    for (int i=0; i<10; i++) {
final int value = i;
Runnable runnable = new Runnable() {
  public void run() {
    System.out.println(value);
  }
};
    }
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
}
```

The problem here is that we have to make a new local data member each time we go through the for loop, so a thought I had today while coding, was to change the above code to the following:

```
public class Access3 {
  public void f() {
    Runnable[] runners = new Runnable[10];
   for (final int[] i={0}; i[0]<runners.length; i[0]++) {
runners[i[0]] = new Runnable() {
  private int counter = i[0];
  public void run() {
    System.out.println(counter);
  }
};
    }
   for (int i=0; i<runners.length; i++)
runners[i].run();
  }
  public static void main(String[] args) {
    new Access3().f();
  }
}
```

We now don't have to declare an additional final local variable. In fact, is it not perhaps true that int[] i is like a common C pointer to an int? It took me 4 years to see this, but I'd like to hear from you if you have heard this idea somewhere else.

I always appreciate any feedback, both positive and negative, so please keep sending your ideas and suggestions. Please also remember to take the time to send this newsletter to others who are interested in Java.

Heinz

Errata

In newsletter 23, I relied on hearsay for some information without checking it out myself. Yes, the amount of memory used by each Thread for its stack was incorrect. According to new measurements, it seems that each stack takes up approximately 20KB, not the 2MB stated in the original newsletter, so for 10000 threads we will need about 200MB. Most operating systems cannot handle that many threads very well anyway, so we want to avoid creating that many. Thanks to Josh Rehman for pointing out this mistake.

It seems that with JDK 1.1.8 each thread takes up 145KB, I'm not sure whether the stacks grow dynamically. Under the JDK 1.2.2 that comes with JBuilder 3, the stacks grow, so I managed to have 100 threads take up roughly 100 MB of memory, or 1MB each. After 1 MB is used up, the VM mysteriously returns without any message so I had to experiment a bit to get this information. Under JDK 1.3.0 I got DrWatsons when I tried to make the stack of each thread grow to any real size. It is unrealistic to expect our program to have stack depths of 10000 method calls, so we could probably quite safely use 50KB as a realistic stack size for each thread.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 026] - Package Versioning

Author: Herman Lintvelt

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 26th issue of "The Java(tm) Specialists' Newsletter". My silence these past two weeks was due to having my hands full moving my outlook contacts list to the new mailserver. Yes, it has finally happened! In addition, I've done some cosmetic changes to the archive of newsletters and as you can see, the newsletter is now sent out in HTML format with syntax highlighted code. I want to thank Peter Carruthers for letting me use some ideas from his newsletter for this new format.

I am very grateful to fellow Java guru Herman Lintvelt for authoring this week's newsletter. He is the only person I know who understands the javax.swing.JTree, perhaps I should call him Dr. Swing. In case you need some serious Java GUI development done, please send him an email, or if you have GUI code written by amateurs that needs to be cleaned up.

Herman did his Bachelor of Science Degree in Computer Science in 1998 at "University" of Stellenbosch, and since then has worked almost exclusively in Java, especially using Swing and JDBC, with bits of JAI and JMF. After only 2 years as sucker^ H^ H^ H^ H^ H^ Hemployee, he decided it's not for him and recently started Polymorph Systems to enter the "freelance contractor" market.

Please remember to forward this newsletter to as many Java enthusiasts as you know who might be interested in receiving such a newsletter. Special thanks to the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

person who sent the last newsletter to their local JUG, it caused about 30 subscriptions :-).

And now for Herman's first newsletter...

Package Versioning

While having a (rare) bit of idle time in this coldest, wettest winter of probably the last decade down here at the Southern tip of Africa, I thought it might be good to write about package versioning. Talking of wettest, I live on a wine-farm near Stellenbosch, and were it not for my trusty old '76 Jeep, I would have been cut off from civilization last week, as the dirt road was turned into a muddy river, but this is not a 4x4 newsletter, so back to Java(tm)...

Versioning of packages is a very good discipline to follow. Yes, if you have source-control package like SourceSafe, CVS, Clearcase, etc, you can (probably) be rest assured that you can keep track of your sourcecode versions. But what happens if your jar-files are finally (after months of intensive testing by expert testers ;-) distributed to the client and the *totally unexpected* happens: a bug shows up! You need to be able to determine whether the client had an old version of the package, as the newest version definitely fixed all the remaining bugs.

I like simple things (that's why this newsletter is not complex), and with package versioning it must be the same: it must be easy and fast to keep the package version information up to date (else I will be too tempted to ignore the versioning process). In a project I worked on previously, we added a class to each java package which implemented the same interface and returned the major, minor, patch and build numbers as hard-coded strings. Added to this was a "PackageVersionInterrogator" class that would then get these strings from all the packages when the debug logger required them. This had a few disadvantages:

the debug logger had to know all the packages for which it had to get the package versions (thus any packages from where a class might *possibly* be loaded, would have to be included in the list of known packages for the debug logger)
if the Java app could not even start up, then the client could not browse the jar-file for versioning information, as the version info was contained in class files.

But Alas!!, once again our friends at Sun had a lot of foresight. A while back I "discovered" the existence of the java.lang.Package class, and what's more: this class contains all kinds of nice accessor methods for versioning information! How do you use it? Easy, for every jar file in your project (where every jar file can contain one or more java packages), you need to create your own manifest.mf file (or whatever name you give to it), and invoke the option of the jar-tool to use this manifest file.

Let's look at how this manifest file will look. Suppose we have some classes in a package called com.worldsgreatestApps.java.media and some in a package called com.worldsgreatestApps.java.media.ui. When we run the jar-tool to create a jar-file

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

for these classes, we use the following manifest.mf file (which will be put in the \META-INF subdir inside the jar file):

```
Manifest-Version: 1.0
Main-Class: jmftest.PackageTest
Specification-Title: Java Media Framework
Specification-Version: 2.0
Specification-Vendor: Sun

Name: com/worldsgreatestApps/java/media/
Implementation-Title: WorldsGreatestApps Super Media Framework
Implementation-Version: 1.0.2
Implementation-Vendor: Polymorph Systems

Name: com/worldsgreatestApps/java/media/ui
Specification-Title: Java Media Framework
Specification-Version: 2.0
Specification-Vendor: Sun
Implementation-Title: WorldsGreatestApps Super Media Framework UI classes
Implementation-Version: 1.1.1
Implementation-Vendor: Polymorph Systems
```

Some observations:

`Main-Class` is not part of this versioning, it indicates the main class to run in the jar file, and thus enables you to "run" this jar file directly by using `java -jar jarfilename.jar`.
`Specification-[Title/Version/Vendor]` specifies information about the specification implemented inside the package(s)
`Implementation-[Title/Version/Vendor]` specifies information about the specific implementation contained in the package.
`Name` (and the empty line in front of it) specifies the name of a package, and must preceed the section for every package in the manifest file. Thus all packages in the jar-file for which we need versioning info (should be all the packages) will have a section in the manifest file.
Both the `Specification` and `Implementation` entries can be specified in the "general" section of the manifest file (like the first `Specification` entries in the example), and these entries will then be used to describe packages that do not have specific entries in the manifest file.
The *`X`*`-Version` entry values are -by convention- in the form of *major.minor.micro(/patch)*, but it can basically be any descriptive string.
The last line of the manifest file should be ended with a newline, else the last line will not be read in probably by the classloader.

How do I use this information in my java code? Well, if the class loader finds package information like this for the package that a class belongs to, it will create a java.lang.Package object for that class, which you can get a handle to by calling on your classinstance.getClass().getPackage(). This object has some methods to get all the info specified in the manifest file.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

If no information is specified for a specific package, the classloader first looks to see if there is a "general section" for the version information (i.e. Specification and Implementation entries at the top of the file, before the first empty line). If it is there, it will use this info to populate the Package object; else it will create a Package object that will return null on the version-info method calls.

What's the advantages of using this method to do your package versioning? Well, it's simple and easy to use. Your debug logger can use the standard Java(tm) API to get package version information, and can get a handle on all the loaded packages by calling Package.getPackages(), for example:

```
/** simple class to list all loaded packages with */
public class PackageLister {
  //...
  public static void listPackages() {
    Package[] allPackages = Package.getPackages();
     System.out.println("All loaded packages:");
    for(int i=0; i < allPackages.length; i++) {
          System.out.println("" + (i+1) +
            ": " + allPackages[i].getName() +
            ": " + allPackages[i].getImplementationTitle() +
            ", version: " + allPackages[i].getImplementationVersion());
    }
  }
  //...
}
```

Another big advantage: on the client site it's easy to open the jar file and view the manifest file manually if (by some operating system or network error of course) your application won't even start up enough to produce debug output.

I hope this inspired those *few* of you who are not using package versioning techniques to start doing it, and those who are using overly complex techniques or tools, to simplify. I'm very interested to know what versioning "systems" are being used out there, so please feel free to email me at herman@javaspecialists.co.za with your ideas and descriptions of the systems you are using for versioning your projects.

Vriendelike groete

Herman

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

..... [Issue 027] - Circular Array List

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 27th issue of "The Java(tm) Specialists' Newsletter". Last week I was talking to a friend about keyboards and he mentioned what a pity it was that we had grown up on QWERTY, when Dvorak is sooo much better for the fingers. I'm sure you've heard the story that QWERTY was designed to slow you down? He also mentioned that some students he studied with removed all the keys from their keyboards and arranged them in Dvorak order. "What a good idea!", I thought, promptly removing all the keys from my notebook and dropping them down in Dvorak order. There are two problems, however:

I touch type QWERTY, so until I find the time to learn Dvorak I want to continue QWERTYing. Some Java applications (JBuilder4, JProbe, etc.) have the extremely annoying habit of changing my input locale to the bottom selection, in my case Dvorak.
Three of the keys are different sizes to the rest of the keys, so in order to remember what they are supposed to be, I stuck stickers on them and the places where they are supposed to be, and wrote on them the appropriate Dvorak keys. Whenever someone sees my notebook now, they ask: "Why do you have stickers on your keys?" They don't see that ALL the keys are in the wrong place! Sometimes we are like that too when it comes to Java. We say: "Java is slow!" In the meantime, we use the wrong collections, we create too many objects, we use += with Strings and print `new java.util.Date()` to the console in a tight loop. When people make statements like "Java is a memory hog" they are demonstrating that they are not looking at the whole picture.

One of the advantages in writing this newsletter is that I get some excellent feedback on some of the things I write about. In newsletter 24 I spoke about self-tuning FIFO queues, the purpose of which was to demonstrate some ideas on self-adapting code, rather than propose a good FIFO implementation. Ecce Jezuch from Poland suggested that I write a circular Array list instead of this self-adapting silly nonsense code, and while we were debating what it would take to give up his summer weather to write such a construct, I received sample code of a partial implementation from Allard Siemelink, an independent Java contractor in the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Netherlands. It is quite simple writing a basic circular array FIFO queue, but I didn't think a basic solution would be acceptable to my readers.

Circular Array List

In this newsletter I present an implementation of java.util.List, which I call the CircularArrayList. It can be used exactly as an ArrayList, i.e. you can remove elements from anywhere in the list, you can add elements to the list, get the size, convert it to an array with the toArray method etc. Some of the methods I've left as an exercise to the reader.

So, the question I didn't answer in newsletter 24 was why java.util.ArrayList is so bad as a FIFO queue when the list contains a lot of elements. I assumed that anybody who had subscribed to an **Advanced** Java(tm) newsletter had at least read the source code of that class and already knew the answer. I think my assumption is probably correct, but here is the source of trouble anyway:

```
public Object remove(int index) {
  RangeCheck(index);

 modCount++;
  Object oldValue = elementData[index];

  int numMoved = size - index - 1;
  if (numMoved > 0)
    System.arraycopy(elementData, index+1, elementData, index,
                     numMoved);
   elementData[--size] = null; // Let gc do its work
    return oldValue;
}
```

In a FIFO queue, we are adding to the back and removing from the front, so if we call remove(0) on a list of 100 elements, numMoved will equal 99, which will cause a call to System.arraycopy(elementData, 1, elementData, 0, 99);. This is the reason why the ArrayList deteriorates as a FIFO queue when it reaches a certain length.

What follows it a CircularArrayList that looks like the ArrayList except that it has a head and a tail that show the first and last elements in the Object element array. In the ArrayList, the head would always be 0. I've written some comments before each method so you can figure out what it does. I recommend that you read the source code of java.util.ArrayList before reading this code.

```
// CircularArrayList.java
import java.util.*;
import java.io.*;

public class CircularArrayList extends AbstractList
  implements List, Serializable
{
  private Object[] elementData;
  // head points to the first logical element in the array, and
  // tail points to the element following the last.  This means
  // that the list is empty when head == tail.  It also means
  // that the elementData array has to have an extra space in it.
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  private int head=0, tail=0;
  // Strictly speaking, we don't need to keep a handle to size,
  // as it can be calculated programmatically, but keeping it
  // makes the algorithms faster.
  private int size=0;

  public CircularArrayList() {
    this(10);
  }

  public CircularArrayList(int size) {
    elementData = new Object[size];
  }

  public CircularArrayList(Collection c) {
    tail = c.size();
    elementData = new Object[c.size()];
    c.toArray(elementData);
  }

  // The convert() method takes a logical index (as if head was
  // always 0) and calculates the index within elementData
  private int convert(int index) {
    return (index + head) % elementData.length;
  }

  public boolean isEmpty() {
    return head == tail; // or size == 0
  }

  // We use this method to ensure that the capacity of the
  // list will suffice for the number of elements we want to
  // insert.  If it is too small, we make a new, bigger array
  // and copy the old elements in.
  public void ensureCapacity(int minCapacity) {
    int oldCapacity = elementData.length;
    if (minCapacity > oldCapacity) {
      int newCapacity = (oldCapacity * 3)/2 + 1;
      if (newCapacity < minCapacity)
         newCapacity = minCapacity;
      Object newData[] = new Object[newCapacity];
      toArray(newData);
      tail = size;
      head = 0;
      elementData = newData;
    }
  }

  public int size() {
    // the size can also be worked out each time as:
    // (tail + elementData.length - head) % elementData.length
    return size;
  }

  public boolean contains(Object elem) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    return indexOf(elem) >= 0;
  }

  public int indexOf(Object elem) {
    if (elem == null) {
      for (int i = 0; i < size; i++)
        if (elementData[convert(i)]==null)
          return i;
    } else {
      for (int i = 0; i < size; i++)
        if (elem.equals(elementData[convert(i)]))
          return i;
    }
    return -1;
  }

  public int lastIndexOf(Object elem) {
    if (elem == null) {
      for (int i = size-1; i >= 0; i--)
        if (elementData[convert(i)]==null)
          return i;
    } else {
      for (int i = size-1; i >= 0; i--)
        if (elem.equals(elementData[convert(i)]))
          return i;
    }
    return -1;
  }

  public Object[] toArray() {
    return toArray(new Object[size]);
  }

  public Object[] toArray(Object a[]) {
    if (a.length < size)
      a = (Object[])java.lang.reflect.Array.newInstance(
        a.getClass().getComponentType(), size);
    if (head < tail) {
      System.arraycopy(elementData, head, a, 0, tail-head);
    } else {
      System.arraycopy(elementData, head, a, 0,
        elementData.length-head);
      System.arraycopy(elementData, 0, a, elementData.length-head,
        tail);
    }
    if (a.length > size)
      a[size] = null;
    return a;
  }

  private void rangeCheck(int index) {
    if (index >= size || index < 0)
      throw new IndexOutOfBoundsException(
        "Index: "+index+", Size: "+size);
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public Object get(int index) {
  rangeCheck(index);
  return elementData[convert(index)];
}

public Object set(int index, Object element) {
  modCount++;
  rangeCheck(index);
  Object oldValue = elementData[convert(index)];
  elementData[convert(index)] = element;
  return oldValue;
}

public boolean add(Object o) {
  modCount++;
  // We have to have at least one empty space
  ensureCapacity(size + 1 + 1);
  elementData[tail] = o;
  tail = (tail+1)%elementData.length;
  size++;
  return true;
}

// This method is the main reason we re-wrote the class.
// It is optimized for removing first and last elements
// but also allows you to remove in the middle of the list.
public Object remove(int index) {
  modCount++;
  rangeCheck(index);
  int pos = convert(index);
  // an interesting application of try/finally is to avoid
  // having to use local variables
  try {
    return elementData[pos];
  } finally {
    elementData[pos] = null; // Let gc do its work
    // optimized for FIFO access, i.e. adding to back and
    // removing from front
    if (pos == head) {
      head = (head+1)%elementData.length;
    } else if (pos == tail) {
      tail = (tail-1+elementData.length)%elementData.length;
    } else {
      if (pos > head && pos > tail) { // tail/head/pos
        System.arraycopy(elementData, head, elementData, head+1,
          pos-head);
        head = (head+1)%elementData.length;
      } else {
        System.arraycopy(elementData, pos+1, elementData, pos,
          tail-pos-1);
        tail = (tail-1+elementData.length)%elementData.length;
      }
    }
    size--;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
  }

  public void clear() {
    modCount++;
    // Let gc do its work
    for (int i=head; i!=tail; i=(i+1)%elementData.length)
      elementData[i] = null;
    head = tail = size = 0;
  }

  public boolean addAll(Collection c) {
    modCount++;
    int numNew = c.size();
    // We have to have at least one empty space
    ensureCapacity(size + numNew + 1);
    Iterator e = c.iterator();
    for (int i=0; i < numNew; i++) {
      elementData[tail] = e.next();
      tail = (tail+1)%elementData.length;
      size++;
    }
    return numNew != 0;
  }

  public void add(int index, Object element) {
    throw new UnsupportedOperationException(
      "This method left as an exercise to the reader ;-)");
  }

  public boolean addAll(int index, Collection c) {
    throw new UnsupportedOperationException(
      "This method left as an exercise to the reader ;-)");
  }

  private synchronized void writeObject(ObjectOutputStream s)
      throws IOException {
    s.writeInt(size);
    for (int i=head; i!=tail; i = (i+1)%elementData.length)
      s.writeObject(elementData[i]);
  }

  private synchronized void readObject(ObjectInputStream s)
      throws IOException, ClassNotFoundException {
    // Read in size of list and allocate array
    head = 0;
    size = tail = s.readInt();
    elementData = new Object[tail];
    // Read in all elements in the proper order.
    for (int i=0; i < tail; i++)
      elementData[i] = s.readObject();
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

There are some bugs in the code above. Please don't ask me what they are; obviously I don't know what the bugs are, else they would not be there anymore. If you do find a bug, please tell me so that I can send an errata next week. I've tried to test the CircularArrayList against an ordinary ArrayList for correctness, and against an ArrayList and a LinkedList for performance.

```
// ArrayListTest.java
import java.util.List;
import java.util.ArrayList;
import java.util.LinkedList;

public class ArrayListTest {
  public static String testList(List list) {
    StringBuffer result = new StringBuffer(1024);
    list.add("ABCD");
    list.add("EFGH");
    list.add("IJKL");
    list.addAll(list);
    result.append(list);
    result.append(list.contains("IJKL"));
    result.append(list.containsAll(new ArrayList()
     {{add("ABCD");add("EFGH");}}));
    result.append(list.equals(new ArrayList(list)));
    for (int i=0; i<6; i++)
      result.append(list.get(i));
    result.append(list.indexOf("EFGH"));
    result.append(list.isEmpty());
    result.append(list.lastIndexOf("EFGH"));
    for (int i=0; i<3; i++) result.append(list.remove(3));
    for (int i=0; i<3; i++) result.append(list.remove(0));
    for (int i=0; i<6; i++) list.add(Integer.toString(i));
    for (int i=0; i<6; i++) result.append(list.get(i));
    Object[] els = list.toArray();
    for (int i=0; i<els.length; i++) result.append(els[i]);
    String[] strs = (String[])list.toArray(new String[0]);
    for (int i=0; i<strs.length; i++) result.append(strs[i]);
    for (int i=0; i<32; i++) {
      list.add(Integer.toHexString(i));
      result.append(list.remove(0));
    }
    result.append(list);
    return result.toString();
  }
  public static void testPerformance(List list, int length) {
    Object job = new Object();
    int iterations = 0;
    for (int j=0; j<length; j++) list.add(job);
    long time = -System.currentTimeMillis();
    while(time + System.currentTimeMillis() < 2000) {
      iterations++;
      for (int j=0; j<100; j++) {
        list.remove(0);
        list.add(job);
      }
    }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    time += System.currentTimeMillis();
    System.out.println(list.getClass() + " managed " +
      iterations + " iterations in " + time + "ms");
  }
  public static void testCorrectness() {
    String al = testList(new ArrayList(6));
    String cal = testList(new CircularArrayList(6));
    if (al.equals(cal)) System.out.println("Correctness Passed");
    else {
      System.out.println("Expected:");
      System.out.println(al);
      System.out.println("But got:");
      System.out.println(cal);
    }
  }
  public static void testPerformance(int length) {
    System.out.println("Performance with queue length = " + length);
    testPerformance(new ArrayList(), length);
    testPerformance(new LinkedList(), length);
    testPerformance(new CircularArrayList(), length);
  }
  public static void main(String[] args) {
    testCorrectness();
    testPerformance(1);
    testPerformance(10);
    testPerformance(100);
    testPerformance(1000);
    testPerformance(10000);
    testPerformance(100000);
  }
}
```

I've tried this out on my trusty notebook with the stickers on the keys. I again count iterations rather than amount of time it takes, as that gives me far more accurate figures. The ArrayList beats our CircularArrayList if there is only one element on the list, since ArrayList in that case does not have to do any copying of elements with System.arraycopy(). The CircularArrayList has a few overheads more than the ArrayList, which make it marginally slower in that one case. Besides that one case, CircularArrayList beats the stuffing out of the other lists for FIFO queue behaviour, AND it can behave as an ordinary ArrayList as well. This makes me wonder why Sun Microsystems didn't implement ArrayList as a CircularArrayList in the first place? I wish I knew the answer to that.

```
Correctness Passed
Performance with queue length = 1
class java.util.ArrayList managed 145246 iterations in 2003ms
class java.util.LinkedList managed 42385 iterations in 2003ms
class CircularArrayList managed 51058 iterations in 2013ms
Performance with queue length = 10
class java.util.ArrayList managed 51236 iterations in 2003ms
class java.util.LinkedList managed 41188 iterations in 2003ms
class CircularArrayList managed 54124 iterations in 2003ms
Performance with queue length = 100
class java.util.ArrayList managed 27019 iterations in 2003ms
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
class java.util.LinkedList managed 39529 iterations in 2003ms
class CircularArrayList managed 52938 iterations in 2004ms
Performance with queue length = 1000
class java.util.ArrayList managed 5300 iterations in 2003ms
class java.util.LinkedList managed 39132 iterations in 2003ms
class CircularArrayList managed 50893 iterations in 2003ms
Performance with queue length = 10000
class java.util.ArrayList managed 535 iterations in 2003ms
class java.util.LinkedList managed 13102 iterations in 2003ms
class CircularArrayList managed 52030 iterations in 2003ms
Performance with queue length = 100000
class java.util.ArrayList managed 9 iterations in 2063ms
class java.util.LinkedList managed 8002 iterations in 2003ms
class CircularArrayList managed 50803 iterations in 2003ms
```

There you go, I hope you bothered to read and try to understand my code. Please remember to forward this newsletter to interested parties.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 028] - Multicasting in Java

Author: Paul van Spronsen

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

This week we again have a guest author, Paul van Spronsen who owns Blue Label Software in South Africa, a small software development company. I have written a

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

lot of code with Paul and he is the best Java programmer that I've had the pleasure of working with. So, if your project needs some extra horse power (ok, brain power ;-), send him an email and I'm sure his company will be able to help you.

A special word of thanks also to Robin Beetge for hacking together a dirty Perl script to generate XML tags for the syntax highlighting. Your little script is saving me a lot of time :)

Multicasting in Java

1. Introduction

This article deals primarily with the subject of multicast communication in Java. I have, however, included some background information to refresh the memory of those who have forgotten how much they know about data communications. If the concepts "datagram", "IP fragment", "reliable protocol" or "multicast" are not clear to you, try referring to the appendices. If the appendices appear shrouded in mystery, go back to your data comms lecturer and demand a refund.

2. Sending multicast datagrams

In order to send any kind of datagram in Java, be it unicast, broadcast or multicast, one needs a java.net.DatagramSocket:

```
DatagramSocket socket = new DatagramSocket();
```

One can optionally supply a local port to the DatagramSocket constructor to which the socket must bind. This is only necessary if one needs other parties to be able to reach us at a specific port. A third constructor takes the local port AND the local IP address to which to bind. This is used (rarely) with multi-homed hosts where it is important on which network adapter the traffic is received. Neither of these is necessary for this example.

This sample code creates the socket and a datagram to send and then simply sends the same datagram every second:

```
DatagramSocket socket = new DatagramSocket();

byte[] b = new byte[DGRAM_LENGTH];
DatagramPacket dgram;

dgram = new DatagramPacket(b, b.length,
  InetAddress.getByName(MCAST_ADDR), DEST_PORT);

System.err.println("Sending " + b.length + " bytes to " +
  dgram.getAddress() + ':' + dgram.getPort());
while(true) {
  System.err.print(".");
  socket.send(dgram);
  Thread.sleep(1000);
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Valid values for the constants are:

`DGRAM_LENGTH`: anything from 0 to 65507 (see section 5), eg 32
`MCAST_ADDR`: any class D address (see appendix D), eg 235.1.1.1
`DEST_PORT`: an unsigned 16-bit integer, eg. 7777

It is important to note the following points:

DatagramPacket does not make a copy of the byte-array given to it, so any change to the byte-array before the `socket.send()` will reflect in the data actually sent;
One can send the same DatagramPacket to several different destinations by changing the address and or port using the `setAddress()` and `setPort()` methods;
One can send different data to the same destination by changing the byte array referred to using `setData()` and `setLength()` or by changing the contents of the byte array the DatagramPacket is referring to;
One can send a subset of the data in the byte array by manipulating offset and length through the `setOffset()` and `setLength()` methods.

3. Receiving multicast datagrams

One can use a normal DatagramSocket to send and receive unicast and broadcast datagrams and to send multicast datagrams as seen in the section 2. In order to receive multicast datagrams, however, one needs a MulticastSocket. The reason for this is simple, additional work needs to be done to control and receive multicast traffic by all the protocol layers below UDP.

The example given below, opens a multicast socket, binds it to a specific port and joins a specific multicast group:

```
byte[] b = new byte[BUFFER_LENGTH];
DatagramPacket dgram = new DatagramPacket(b, b.length);
MulticastSocket socket =
  new MulticastSocket(DEST_PORT); // must bind receive side
socket.joinGroup(InetAddress.getByName(MCAST_ADDR));

while(true) {
  socket.receive(dgram); // blocks until a datagram is received
  System.err.println("Received " + dgram.getLength() +
    " bytes from " + dgram.getAddress());
  dgram.setLength(b.length); // must reset length field!
}
```

Values for DEST_PORT and MCAST_ADDR must match those in the sending code for the listener to receive the datagrams sent there. BUFFER_LENGTH should be at least as long as the data we intend to receive. If BUFFER_LENGTH is shorter, the data will be **truncated silently** and dgram.getLength() will return b.length.

The MulticastSocket.joinGroup() method causes the lower protocol layers to be informed that we are interested in multicast traffic to a particular group address. One may execute joinGroup() many times to subscribe to different groups. If multiple MulticastSockets bind to the same port and join the same multicast group, they will all receive copies of multicast traffic sent to that group/port.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

As with the sending side, one can re-use ones DatagramPacket and byte-array instances. The receive() method sets length to the amount of data received, so remember to reset the length field in the DatagramPacket before subsequent receives, otherwise you will be silently truncating all your incoming data to the length of the shortest datagram previously received.

One can set a timeout on the receive() operation using socket.setSoTimeout(timeoutInMilliseconds). If the timeout is reached before a datagram is received, the receive() throws a java.io.InterruptedIOException. The socket is still valid and usable for sending and receiving if this happens.

4. Multicasting and serialization

We have seen in the previous sections that we can multicast anything we can fit into a byte array. Conveniently for us, one of those things is a serialized object.

Object serialization is based on the assumption of a stream (ObjectOutput**Stream**, ObjectInput**Stream**), so we have to do a little massaging to squeeze this into our datagram paradigm. ObjectOutputStream writes a stream header (containing a magic number and version number) to the stream on construction and ObjectInputStream reads and checks this on construction (ever wondered why ObjectInputStream's constructor blocks until the ObjectOutputStream has been constructed on the sending side?). This is the reason one always attaches the ObjectOutputStream to the outgoing side of a socket before attaching the ObjectInputStream to the incoming side.

In order to multicast objects, we need to arrange that the stream header information is in each datagram. The simplest way to ensure this is to create a new ObjectOutputStream for each datagram we send and a new ObjectInputStream for each one we receive. We could probably avoid these instantiations by extending the two classes in question, but I'm not going into that here.

On the sending side, we can do something like this:

```
ByteArrayOutputStream b_out = new ByteArrayOutputStream();
ObjectOutputStream o_out = new ObjectOutputStream(b_out);

o_out.writeObject(new Message());

byte[] b = b_out.toByteArray();

DatagramPacket dgram = new DatagramPacket(b, b.length,
  InetAddress.getByName(MCAST_ADDR), DEST_PORT); // multicast
socket.send(dgram);
```

In addition, on the receiving side we can do something like this:

```
byte[] b = new byte[65535];
ByteArrayInputStream b_in = new ByteArrayInputStream(b);
DatagramPacket dgram = new DatagramPacket(b, b.length);

socket.receive(dgram); // blocks
ObjectInputStream o_in = new ObjectInputStream(b_in);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
Object o = o_in.readObject();
dgram.setLength(b.length); // must reset length field!
b_in.reset(); // reset so next read is from start of byte[] again
```

Note that one can re-use the ByteArray* Streams, byte arrays and DatagramPackets on both sides. Only the Object* Streams need be recreated.

5. Datagram sizes

The IP spec allows for datagrams up to 65535 bytes in length, *including* the IP header. If the underlying protocol layers cannot support this size (Ethernet's MTU is 1500 bytes), IP fragments the datagrams into several smaller datagrams. On the receive side, IP reassembles the datagram before delivering it to higher layer protocols, like UDP. If any of the fragments do not arrive at the destination, the entire datagram is discarded, i.e. there is no partial delivery of IP and therefore UDP datagrams.

Since the normal IP header is 20 bytes long and the UDP header is always 8 bytes long, one would expect the maximum UDP data length to be 65535-8-20 = 65507. Somehow, however, the combination of Win2k and JDK1.3.1 manages to successfully send as much as 65527 bytes per datagram. I would be interested to hear whether users of a real operating system experienced the same.

It is *very important* to note that although the IP spec allows for datagrams up to 65535 bytes, it only requires implementations to support up to 576 byte IP datagrams including IP and higher protocol headers. Since the maximum IP header length is 64 and the UDP header length is 8, it is safe to send up to 504 byte UDP datagrams and ***expect*** the receiving side to handle it (yes, even your Palm Pilot if it has a TCP/IP stack). I have not come across a full size (i.e. non-embedded) system that cannot handle the full 64k-1, though.

6. Effect of fault conditions

UDP does not gaurantee delivery or notification of non-delivery. If you send a unicast packet to a host that does not exist, is down or is not listening on that port, you will not know about it. If you send a broadcast or multicast packet and nobody receives it or is even listening, you will not know about it.

On Win2k the network adapter settings are reset if it is detected that the link is not available. With Ethernet, for example, if you unplug the LAN cable so that there is no link available, Win2K detects this and effectively shuts down the adapter at the IP level. It clears its IP address and will not attempt to use it. The effect of this is that sockets cannot bind to a port, so all new *Socket calls fail. Sockets that are already created function correctly if you unplug and replug the cable.

On my notebook, local communication (sender and listener on the same machine) began to fail when I unplugged the LAN cable. It gets nastier than this:- a listener started before I unplugged the cable could not hear traffic from a sender started after I had plugged the cable back in. But wait, there's more! I started another listener after the cable was back in and it *and the listeners started before I unplugged the cable*, all receive the multicasts again.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

On WinNT4, my experience has been that the adapter is not "shutdown" when the cable is unplugged and one does not have these weird effects.

7. Multiple listeners and unicast packets

Since one can send unicast packets using the same MulticastSocket instance as for ones multicasts, it makes sense to mention how unicasts are handled when there is more than one listener, which can only be when they are all on the same machine.

Unicast traffic sent to the port will be received by only one of the listeners with a socket bound to the port. With my test setup, the last socket to bind to the port receives the unicast traffic. On WinNT4, the first one to bind receives it. I don't know of any rules covering how unicast traffic should be handled in the case of multiple listeners, so don't rely on it being handled in any particular way.

8. Further reading

See the RFCs for IP(791), UDP(768) and IP multicasting(1112). Compared to some of the ISO and IEEE stuff I've seen, they're recreational reading material.

APPENDICES

A. Protocol "reliability"

You may have heard TCP described as a "reliable" protocol and UDP as an "unreliable" protocol. It is easy, but dangerous, to jump to conclusions about what this means. Being "reliable" does not mean that TCP will deliver your data under all circumstances (try unplugging the LAN cable for a day and see). Being "unreliable", does not mean UDP will arbitrarily throw away your data. "Unreliable" is a loaded term and I prefer to use "non-reliable" which indicates more that it lacks the gaurantees of a "reliable" protocol, rather than labelling it as some sort of untrustworthy servant.

Enough about what reliability, or lack of it, does not mean. A "reliable" protocol like TCP guarantees that it will deliver your data correctly and in order of transmission *or inform you that it could not*.

A "non-reliable" protocol, like UDP, does what is called "best-effort delivery". Essentially, given enough available resources (buffers, bandwidth etc) UDP will deliver your data correctly. It will not deliver incorrect data, but it could deliver data in a different order to which it was sent or not at all.

The NFS (Network File System) protocol uses UDP to communicate between the server and the client. IMHO, this is a testament to the "reliability" of UDP as a transport. Of course, NFS implements its own reliability mechanisms (timeouts and retransmissions) on top of UDP to be sure.

B. Stream vs Datagrams

The differences between TCP and UDP don't end with reliability. They are fundamentally different in their data model. TCP is stream based and UDP is datagram based. This means that with UDP, if data is lost or delivered out of order, it happens with datagram granularity.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Since TCP is stream based, it does not honour your message boundaries. If you implement your own message passing system using TCP, you will find that doing a send() call of n bytes on one side of the connection does not necessarily result in n bytes being returned by the "corresponding" read() call on the other side. TCP rides on top of IP, which is datagram based, so there is packetizing happening when TCP data is sent, but TCP is at liberty to split your send() up into several actual packets or to coalesce several send() operations into one packet.

C. nCasting

In the case of TCP, the number of intended recipients of transmitted data is always exactly one (like a telephone call). In general, this is not the case. Everybody is aware of broadcast communication (like radio or television) where there is one sender and any number of recipients. As most people know the same exists in data communications.

Broadcast communication is frowned upon by network admins because they spend a huge portion of their budget trying to provide bandwidth using network switches, only to have this all defeated by broadcast traffic being delivered to every segment of their LANs. Broadcast communication also causes an interrupt and the associated processing on every node on the connected LAN, always. Ones Ethernet hardware, for example, cannot determine whether the host is interested in any particular broadcast packet and must therefore deliver the packet to the upper protocol layers to make the decision. This is the reason Doom 1.1 network games were banned on many LANs. The number of broadcasts used caused such high interrupt processing loads on all the hosts on networks where it was played. Thankfully, Doom 1.2 came along to avert boredom during my time at university.

Where broadcasting is a mechanism intended to deliver data to all hosts on a network or subnetwork, multicasting is a mechanism to deliver data to a group of interested hosts on a network. Many network adapters provide some sort of rudimentary multicast filtering. In many cases, a host not interested in a particular multicast group will not even be interrupted by its network hardware.

In the TCP/IP protocol family, UDP is used for broadcast and multicast (and some unicast) traffic. As a result, broadcast and multicast traffic is datagram based and non-reliable.

Reliability, datagram vs stream based and unicast vs multicast/broadcast traffic are all orthogonal concepts. It is not inconceivable to have a reliable, stream based multicast protocol, or any other combination of those features.

D. IP Multicast addresses

All class D IP addresses are multicast addresses. Class D IP addresses are those that begin with 1110, that is, all addresses from 224.0.0.0 to 235.255.255.255. Some are pre-assigned for specific applications, but most are available for forming ad hoc multicast groups. There is a mapping between IP multicast addresses and Ethernet addresses, described in **RFC1112:** *"An IP host group address is mapped to an Ethernet multicast address by placing the low-order 23-bits of the IP address into the low-order 23 bits of the Ethernet multicast address 01-00-5E-00-00-00 (hex). Because there are 28 significant bits in an IP host group address, more than one host group address may map to the same Ethernet multicast address."*

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 029] - Determining Memory Usage in Java

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

A special thanks to Chris Preimesberger of DevX for featuring our SoftHashMap issue of this newsletter on their website.

Welcome to the 29th issue of "The Java(tm) Specialists' Newsletter". I could start off with a witty comment about how the newsletter is going to hit the **big three** at the next issue, but I might step on the toes of my old friend (haha) John Green who is turning 30 today - happy birthday! At least I'm not *that* old yet :-) By the time you read the next newsletter, or maybe this newsletter, I will probably be father the second time round.

This week I am showing you one of my most dear trade secrets. Please be very careful who you show this newsletter to, only send it to friends and people on your local JUG. If this gets into the wrong hands, project troubleshooters like me will be out of a job.

One of the fun parts in Java is guessing how much memory is being used by your object. We are conditioned to ignore memory altogether when programming in Java and that can easily land us in trouble. Java does not have a construct like C/C++ that tells us how much space an object is taking, at least until this newsletter...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Warning: The results in this newsletter were derived experimentally rather than looking at the innards or the JVM. Please try out the experiments if you are running on a non-WinNT machine and tell me if you get different results.

Memory Usage in Java

In Java, memory is allocated in various places such as the stack, heap, etc. In this newsletter I'm only going to look at objects which are stored on the heap. Please don't take me to task for not mentioning the others, they might appear in a future newsletter.

Say I have a class Foo, how much memory will one instance of that class take? The amount of memory can be determined by looking at the data members of the class and all the superclasses' data members. The algorithm I use works as follows:

The class takes up at least 8 bytes. So, if you say `new Object();` you will allocate 8 bytes on the heap.
Each data member takes up 4 bytes, except for long and double which take up 8 bytes. Even if the data member is a byte, it will still take up 4 bytes! In addition, the amount of memory used is increased in 8 byte blocks. So, if you have a class that contains one byte it will take up 8 bytes for the class and 8 bytes for the data, totalling 16 bytes (groan!).
Arrays are a bit more clever, at least smaller primitives get packed. I'll deal with these later.

In order to be able to test many different types of objects, I have written a MemoryTestBench class that takes an ObjectFactory which is able to create the type of object that you want to test. The MemoryTestBench can either tell you how many bytes are used by that object or it can print out a nicely formatted result for you. You get the most accurate results if you make sure that supplementary memory is already allocated when you start counting. I therefore construct the object, call the methods for finding the memory, and then set the handle to null again. The garbage collector is then called many times, which should free up all unused memory. The memory is then counted, the object created, garbage collected, and the memory counted again. The difference is the amount of memory used by your object, voila!

```
public class MemoryTestBench {
  public long calculateMemoryUsage(ObjectFactory factory) {
    Object handle = factory.makeObject();
    long mem0 = Runtime.getRuntime().totalMemory() -
      Runtime.getRuntime().freeMemory();
    long mem1 = Runtime.getRuntime().totalMemory() -
      Runtime.getRuntime().freeMemory();
    handle = null;
    System.gc(); System.gc(); System.gc(); System.gc();
    System.gc(); System.gc(); System.gc(); System.gc();
    System.gc(); System.gc(); System.gc(); System.gc();
    System.gc(); System.gc(); System.gc(); System.gc();
    mem0 = Runtime.getRuntime().totalMemory() -
      Runtime.getRuntime().freeMemory();
    handle = factory.makeObject();
    System.gc(); System.gc(); System.gc(); System.gc();
    System.gc(); System.gc(); System.gc(); System.gc();
    System.gc(); System.gc(); System.gc(); System.gc();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    System.gc(); System.gc(); System.gc(); System.gc();
    mem1 = Runtime.getRuntime().totalMemory() -
      Runtime.getRuntime().freeMemory();
    return mem1 - mem0;
  }
  public void showMemoryUsage(ObjectFactory factory) {
    long mem = calculateMemoryUsage(factory);
    System.out.println(
      factory.getClass().getName() + " produced " +
      factory.makeObject().getClass().getName() +
      " which took " + mem + " bytes");
  }
}
```

The ObjectFactory interface looks like this:

```
public interface ObjectFactory {
  public Object makeObject();
}
```

Basic Objects

Let's start with the easiest case, a BasicObjectFactory that simply returns a new instance of Object.

```
public class BasicObjectFactory implements ObjectFactory {
  public Object makeObject() {
    return new Object();
  }
}
```

When we run this, we get the following output:

BasicObjectFactory produced java.lang.Object which took 8 bytes

Bytes

I suggested earlier that bytes are *not* packed in Java and that memory usage is increased in 8 byte blocks. I have written the ByteFactory and the ThreeByteFactory to demonstrate this:

```
public class ByteFactory implements ObjectFactory {
  public Object makeObject() {
    return new Byte((byte)33);
  }
}

public class ThreeByteFactory implements ObjectFactory {
  private static class ThreeBytes {
    byte b0, b1, b2;
  }
  public Object makeObject() {
    return new ThreeBytes();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }
}
```

When we run these, we get the following output:

```
ByteFactory produced java.lang.Byte which took 16 bytes
ThreeByteFactory produced ThreeByteFactory$ThreeBytes which took 24 bytes
```

This is great (not). When I first started using Java I used to spend hours deciding whether a variable should be an int or short or a byte in order to minimize the memory footprint. **I was wasting my time.** As I said earlier, I don't know if this is only a problem under NT or if it's the same on all platforms. Knowing Java's dream of being equally inefficient on all platforms, I suspect that it would be the same.

Booleans

Let's carry on and look at a smaller unit of information, the boolean. Now a boolean is simply a bit, true or false, yes or no, zero or one. If I have a class that contains 64 booleans, guess how much memory it will take? 8 for the class, and 4 for each of the boolean data members, i.e. 264 bytes!!! Since a boolean is essentially the same as a bit, we could have stored the same information in one long. If you don't believe me, have a look at the following class:

```
public class SixtyFourBooleanFactory implements ObjectFactory {
  private static class SixtyFourBooleans {
    boolean a0, a1, a2, a3, a4, a5, a6, a7;
    boolean b0, b1, b2, b3, b4, b5, b6, b7;
    boolean c0, c1, c2, c3, c4, c5, c6, c7;
    boolean d0, d1, d2, d3, d4, d5, d6, d7;
    boolean e0, e1, e2, e3, e4, e5, e6, e7;
    boolean f0, f1, f2, f3, f4, f5, f6, f7;
    boolean g0, g1, g2, g3, g4, g5, g6, g7;
    boolean h0, h1, h2, h3, h4, h5, h6, h7;
  }
  public Object makeObject() {
    return new SixtyFourBooleans();
  }
}
```

When we run this, we get the following output:

```
SixtyFourBooleanFactory produced SixtyFourBooleanFactory$SixtyFourBooleans
  which took 264 bytes
```

Admittedly, the example was a little bit contrived, as you would seldom have that many booleans in one class, but I hope you get the idea.

Sun must have realised this problem so they made constants in java.lang.Boolean for TRUE and FALSE that both contain instances of java.lang.Boolean. I think that the constructor for Boolean should have been private to stop people from creating 16 byte objects that are completely unnecessary.

Arrays of Boolean Objects

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

A Boolean Array takes up 16 bytes plus 4 bytes per position with a minimum of 8 bytes at a time. In addition to that, we obviously have to count the actualy space taken by Boolean objects.

```
public class BooleanArrayFactory implements ObjectFactory {
  public Object makeObject() {
    Boolean[] objs = new Boolean[1000];
    for (int i=0; i<objs.length; i++)
      objs[i] = new Boolean(true);
    return objs;
  }
}
```

Try guess how many bytes would be taken up by a Boolean array of size 1000 with Boolean objects stuck in there. Ok, I'll help you: 16 + 4*1000 (for the pointers) + 16*1000 (for the actual Boolean objects) = 20016. Run the code and see if I'm right ;-) If we, instead of making a new Boolean object each time, use the Flyweights provided in Boolean, we'll get to 16 + 4*1000 = 4016 bytes used.

Primitives get packed in arrays, so if you have an array of bytes they will each take up one byte (wow!). The memory usage of course still goes up in 8 byte blocks.

```
public class PrimitiveByteArrayFactory implements ObjectFactory {
  public Object makeObject() {
    return new byte[1000];
  }
}
```

When we run this, we get the following output:

```
PrimitiveByteArrayFactory produced [B which took 1016 bytes

java.lang.String
```

Strings actually fare quite well since they can be "internalised" meaning that only one instance of the same String is kept. If you, however, construct your String dynamically, it will not be interned and will take up a bit of memory. Inside String we find:

```
// ...
private char value[];
private int offset;
private int count;
private int hash = 0;
// ...
```

Say we want to find out how much "Hello World!" would take. We start adding up 8 (for the String class) + 16 (for the char[]) + 12 * 2 (for the characters) + 4 (value) + 4 (offset) + 4 (count) + 4 (hash) = 64 bytes. It's quite difficult to measure this, as we have to make sure the String is not internalized by the JVM. I used the StringBuffer to get this right:

```
public class StringFactory implements ObjectFactory {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public Object makeObject() {
    StringBuffer buf = new StringBuffer(12);
    buf.append("Hello ");
    buf.append("World!");
    return buf.toString();
  }
}
```

When we run this, we get, as expected, the following output:

StringFactory produced java.lang.String which took 64 bytes

java.util.Vector

Now we get to the real challenge: How much does a java.util.Vector use in memory? It's easy to say, now that we have a MemoryTestBench, but it's not so easy to explain. We start by looking inside the java.util.Vector class. Inside we find the following:

```
// ...
protected Object elementData[];
protected int elementCount;
// ...
```

Using the knowledge we already have, we decide that the amount of memory used will be 8 (for the class) + 4 (for the pointer to elementData) + 4 (for elementCount). The elementData array will take 16 (for the elementData class and the length) plus 4 * elementData.length. We then follow the hierarchy up and discover the variable **int** modCount in the superclass java.util.AbstractList, which will take up the minimum 8 bytes. For a Vector of size 10, we will therefore take up: 8 + 4 + 4 + 16 + 4*10 + 8 = 80 bytes, or simply 40 + 4*10 = 80 bytes, which agrees with our experiment:

```
public class VectorFactory implements ObjectFactory {
  public Object makeObject() {
    return new java.util.Vector(10);
  }
}
```

When we run this, we get the following output:

VectorFactory produced java.util.Vector which took 80 bytes

So, what happens when we create a JTable with a DefaultTableModel with 100x100 cells? The DefaultTableModel keeps a Vector of Vectors so this will take 40 + 4*100 + (40 + 4*100) * 100 = 440 + 44000 = 44440 bytes just for the empty table. If we put an Integer in each cell, we will end up with another 100*100*16 = 160'000 bytes used up.

java.util.LinkedList

What's better, a java.util.LinkedList or a java.util.ArrayList? Experienced followers of these newsletters will of course say: "Neither, the CircularArrayList is better" ;-). Let's see what happens when we put 10000 objects into an ArrayList (which uses the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

same amount of memory as the Vector) vs. a LinkedList. Remember that each Object takes up 8 bytes, so we will subtract 80000 bytes from each answer to get comparable values:

```
import java.util.*;
public class FullArrayListFactory implements ObjectFactory {
  public Object makeObject() {
    ArrayList result = new ArrayList(10000);
    for (int i=0; i<10000; i++) {
      result.add(new Object());
    }
    return result;
  }
}
```

```
import java.util.*;
public class FullLinkedListFactory implements ObjectFactory {
  public Object makeObject() {
    LinkedList result = new LinkedList();
    for (int i=0; i<10000; i++) {
      result.add(new Object());
    }
    return result;
  }
}
```

When we run this, we get the following output:

```
FullArrayListFactory produced java.util.ArrayList which took 120040 bytes
FullLinkedListFactory produced java.util.LinkedList which took 320048 bytes
```

When we subtract 80000 bytes from each, we find that the ArrayList takes up 40040 bytes (as expected) and the LinkedList uses 240048 bytes. How many of us consider issues like this when we code?

We have come to the end of yet another newsletter. I am trying to put newsletters together that will be worthwhile to send out, so as a result they will not always appear every week, unless I feel particularly inspired.

Until the next issue...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

..... [Issue 030] - What do you Prefer?

Author: Herman Lintvelt (Polymorph Systems)

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

With the arrival of my second child, a 4.2kg daughter which we called Nicola Constance Bettina Kabutz, I have been rather busy changing nappies, rocking the child to sleep, and more exhaustingly, helping my 3 year old son cope with life in general. Fortunately for my die-hard supporters out there, Herman Lintvelt (herman@javaspecialists.co.za) stepped in and saved the day. I promise to pull my socks up and get these things done more regularly as soon as life returns to stability.

With regards

Heinz

Recently I've downloaded JDK 1.4 beta 2, and then forgot about it for a while as my struggles with JMF required all my resources. But then, on a cold Worcester (South Africa) evening, while sitting in front of my fireplace with a nice warm fire heating the room, I was thinking about things I prefer. My preferences.

Do not worry, I won't carry on being philosophical. It actually reminded me of the new Preferences API in JDK1.4, so I put away the red wine, and pulled out my laptop. And was I pleasantly surprised.

The Preferences API

The guys at Sun has seen the need for handling preferences in a somewhat better and easier to use way than using Properties or implementing a complex preference subsystem that saves it to a database or some other backing store. So they created the Preferences API.

Using this API starts with the Preferences class in the java.util.prefs package. This class represents a preference node in some kind of preference hierarchy. Such a

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

node can have child nodes, as well as key-value pairs belonging to it (similar to Windows Registry). The 4 important static methods of Preferences are:

```
// return system preference-node for package that O belongs to
Preferences systemNodeForPackage(Object 0);
// return root node of system preferences
Preferences systemRoot();
// return system preference-node for package that O belongs to
Preferences userNodeForPackage(Object O);
// return root node of user preferences
Preferences userRoot();
```

Some explanation is probably needed. The preference data gets saved in two tree-like structures in an implementation specific way. The JDK for Windows version saves it in the Windows Registry, but it is possible to create one's own implementation that might for example use a database. The one tree is used to store user-specific preferences (each user on a system will have a seperate tree), and the other tree stores system preferences. (The definition of user and system depends on the preferences implementation. In the Windows JDK version it maps to Windows users and system-wide preferences.) Each node in this tree can be represented by a Preferences object.

However, if you're like me you do not like theory too much (and that's what javadocs are for), so let us explore this API with an example: a "Cross-platform Registry Editor".

Cross-platform Registry Editor

The idea of this Java tool is to be able to view and edit preferences saved via the Preferences API, no matter on what platform it is executed (i.e. the backing store used is transparent to the user).

Preference Nodes

We implement the class PreferencesEditor as a JDialog, and it must contain a JTree to present the preferences trees (user and/or system), and a JTable to display and edit the actual preference values. We need the following inner classes: PrefTreeNode to represent a preference node in the JTree, PrefTableModel to handle the display and editing of preference values in the table, and PrefTreeSelectionListener to update the JTable with the currently selected preference node.

I list the code for PreferenceEditor with discussions in between the code.

```
//add all other necessary imports here
import java.util.prefs.Preferences;
import java.util.prefs.BackingStoreException;

public class PreferencesEditor extends JDialog {
  JTree prefTree;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
 JTable editTable;

 /**
  * Creates PreferencesEditor dialog that show all System and
  * User preferences.
  * @param owner owner JFrame
  * @param title title of dialog
  */
 public PreferencesEditor(JFrame owner, String title){
  this(owner, title, null, true, null, true);
 }

 /**
  * @param owner owner JFrame
  * @param title title of dialog
  * @param userObj the package to which this object belongs is
  * used as the root-node of the User preferences tree (if
  * userObj is null, then the rootnode of all user preferences
  * will be used)
  * @boolean showUserPrefs if true, then show user preferences
  * @param systemObj the package to which this object belongs is
  * used as the root-node of the System preferences tree (if
  * systemObj is null, then the rootnode of all system
  * preferences will be used)
  * @param showSystemPrefs if true, then show system preferences
  */
 public PreferencesEditor(JFrame owner, String title,
    Object userObj, boolean showUserPrefs, Object systemObj,
    boolean showSystemPrefs) {
  super(owner, title);
  getContentPane().setLayout(new BorderLayout(5,5));
  setSize(640,480);
  createTree(userObj, showUserPrefs, systemObj, showSystemPrefs);
  editTable = new JTable();
  createSplitPane();
  createButtonPanel();
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }
```

As mentioned in the code comments, there are two constructors: one give access to all the system and user preferences, and one can be used to only display/edit a specified subset of the preferences. Let's first look at the createTree(...), createUserNode(...) and createSystemRootNode(...) methods to see how this is done:

```
  private void createTree(Object userObj, boolean showUserPrefs, Object
systemObj, boolean showSystemPrefs){
    DefaultMutableTreeNode rootNode = new
DefaultMutableTreeNode("Preferences");
    if (showUserPrefs) {
     rootNode.add(createUserRootNode(userObj));
    }
    if (showSystemPrefs) {
     rootNode.add(createSystemRootNode(systemObj));
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
    DefaultTreeModel model = new DefaultTreeModel(rootNode);
    prefTree = new JTree(model);
    prefTree.addTreeSelectionListener(new PrefTreeSelectionListener());
  }
  private MutableTreeNode createSystemRootNode(Object obj) {
    try {
      PrefTreeNode systemRoot;
      if (obj==null) {
        systemRoot = new PrefTreeNode(Preferences.systemRoot());
      } else {
        systemRoot = new PrefTreeNode(Preferences.systemNodeForPackage(obj));
      }
      return systemRoot;
    } catch (BackingStoreException e) {
      e.printStackTrace();
      return new DefaultMutableTreeNode("No System Preferences!");
    }
  }
  private MutableTreeNode createUserRootNode(Object obj) {
    try {
      PrefTreeNode userRoot;
      if (obj==null) {
        userRoot = new PrefTreeNode(Preferences.userRoot());
      } else {
        userRoot = new PrefTreeNode(Preferences.userNodeForPackage(obj));
      }
      return userRoot;
    } catch (BackingStoreException e) {
      e.printStackTrace();
      return new DefaultMutableTreeNode("No User Preferences!");
    }
  }
```

If the user specify a userObj (and showUserPrefs=true), then Preferences.userNodeForPackage(userObj) gets called in creteUserRootNode. This will return a Preferences object that represents the preferences node that maps to the package structure of userObj. If this preference node does not yet exist in the backing store, it gets created. For example, if I call createUserNode(**new** com.polymorph.MyClass()), then the preference node "com/polymorph" will be returned, and its parent node will be "com" (in the user preference tree). If the user pass **null** as parameter, then Preferences.userRoot() gets called, which return the root node of the user preferences tree (for the current user). The same goes for createSystemRootNode and the system preferences.

Of course we need a way of representing a preference node in a JTree, and this is what the PrefTreeNode inner class is for.

```
  class PrefTreeNode extends DefaultMutableTreeNode {
    Preferences pref;
    String nodeName;
    String[] childrenNames;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    public PrefTreeNode(Preferences pref)  throws BackingStoreException {
      this.pref = pref;
      childrenNames = pref.childrenNames();
    }
    public Preferences getPrefObject(){
      return pref;
    }
    public boolean isLeaf(){
      return ((childrenNames==null)||(childrenNames.length == 0));
    }
    public int getChildCount(){
      return childrenNames.length;
    }
    public TreeNode getChildAt(int childIndex){
      if(childIndex < childrenNames.length){
        try {
          PrefTreeNode child = new
PrefTreeNode(pref.node(childrenNames[childIndex]));
          return child;
        } catch (BackingStoreException e) {
          e.printStackTrace();
          return new DefaultMutableTreeNode("Problem Child!");
        }
      }
      return null;
    }
    public String toString(){
      String name = pref.name();
      if ((name == null)||("".equals(name))){ //if root node
        name = "System Preferences";
        if (pref.isUserNode()) name = "User Preferences";
      }
      return name;
    }
  }
```

This inner class decorates a Preferences object to be used as a MutableTreeNode in a JTree. All the child preferences nodes of this object are accessed via the pref.childrenNames() call, and stored in a String array. This array is then used to calculate the number of children nodes, whether this is a leaf node, etc. getChildAt gets a specific child node, mainly via the pref.node(childrenNames[childIndex]) call. Preferences.node("nodeString") returns a Preferencesobject for the node specified by "nodeString". It can specify a node relative to the current one, ex. "child1", which will return a child preference node (as we use it in getChildAt), or an absolute path can be specified, ex. "/com/polymorph/UI" will return a node "UI", with parent node "polymorph".

Preference Key-Value pairs

OK, our editor is now able to handle the nodes in the user and/or system preferences hierarchy, but how to we actually access the preference values? Well, the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Preferences API allows us to save preferences in our custom defined preferences structure in a very similar way as we would in a Hashmap: we put key-value pairs in the preferences node, where the key is a specific preference setting name, and the value can either be a String, int, long, boolean, float, double or byte[]. Once you have a Preferences object, you can just call put("keyStr", "valueStr"), or putLong("keyStr", 123l), etc. And you can retrieve these values via the get("keyStr", "defaultStr"), or getLong("keyStr", 233 /*defaultVal*/), etc. methods. Note that for every get method, a default value must be supplied. This forces you to think about default values for when the preferences cannot be loaded from the backing store, thus allowing your application to continue even though preferences could not be loaded.

In our editor example, we access these key-value pairs in a JTable, and we need the PrefTableModel to do this:

```
 class PrefTableModel extends AbstractTableModel {
   Preferences pref;
   String[] keys;
   public PrefTableModel(Preferences pref){
     this.pref = pref;
     try{
       keys = pref.keys();
     } catch (BackingStoreException e) {
       System.out.println("Could not get keys for Preference node:
"+pref.name());
       e.printStackTrace();
       keys = new String[0];
     }
   }
   public String getColumnName(int column) {
     switch(column){
     case 0: return "Key";
     case 1: return "Value";
     default: return "-";
     }
   }
   public boolean isCellEditable(int rowIndex, int columnIndex) {
     switch(columnIndex) {
     case 0: return false;
     case 1: return true;
     default: return false;
     }
   }
   public void setValueAt(Object aValue, int rowIndex, int columnIndex) {
     pref.put(keys[rowIndex], aValue.toString());
     try{
       pref.sync(); //make sure the backing store is synchronized with latest update
     } catch (BackingStoreException e) {
       System.out.println("Error synchronizing backStore with updated value");
       e.printStackTrace();
     }
   }
   public Object getValueAt(int row, int column){
     String key = keys[row];
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    if (column==0) return key;
    Object value = pref.get(key, "(Unknown)");
    return value;
  }
  public int getColumnCount(){
    return 2;
  }
  public int getRowCount(){
    return keys.length;
  }
}
```

In the PrefTableModel constructor, pref.keys returns all the key-names stored in the pref node. These key-names are then used in getValueAt to get the value of either a key or value-column cell. If we want the value of a Value-column cell, we get it as a String via the pref.get(key, **"Unknown"**) call (default value="Unknown"), as the Preferences API unfortunately does not seem to allow us to retrieve it as an Object. Thus all values are presented as String in the table, but this should not be a problem, as it seems that these values are saved as Strings anyway in the backing store. getLong, getBoolean, etc. tries and interpret the saved string-value as a **long**, **boolean**, etc.

Only the Value-column cells are editable, and the setValueAt method uses pref.put(key-name, aValue) to update the edited value. It also calls pref.sync() that forces any updates to be synchronized with the backing store.

How do we connect this table model to the preference tree? Well, the PreferencesEditor constructor creates a JTable object (editTable), and then we use the PrefTreeSelectionListener inner class to update the table model of this table.

```
class PrefTreeSelectionListener implements TreeSelectionListener{
  public void valueChanged(TreeSelectionEvent e) {
    try {
      PrefTreeNode node = (PrefTreeNode)e.getPath().getLastPathComponent();
      Preferences pref = node.getPrefObject();
      editTable.setModel(new PrefTableModel(pref));
    } catch (ClassCastException ce) {
      System.out.println("Node not PrefTreeNode!");
      editTable.setModel(new DefaultTableModel());
    }
  }
}
```

The createTree method adds an instance of PrefTreeSelectionListener to the JTree as a listener.

All that now remains to be defined are the createSplitPane() and createButtonPanel methods, and none of them contains any surprises:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  private void createSplitPane(){
    JSplitPane splitPane = new JSplitPane();
    splitPane.setOrientation(JSplitPane.HORIZONTAL_SPLIT);
    splitPane.setOneTouchExpandable(true);
    splitPane.setLeftComponent(new JScrollPane(prefTree));
    splitPane.setRightComponent(new JScrollPane(editTable));
    getContentPane().add(splitPane, BorderLayout.CENTER);
  }
  private void createButtonPanel(){
    JPanel buttonPanel = new JPanel(new BorderLayout(5,5));
    JButton closeButton = new JButton("Close");
    closeButton.addActionListener(new ActionListener(){
      public void actionPerformed(ActionEvent e) {
        System.exit(0);
      }
    });
    buttonPanel.add(closeButton, BorderLayout.EAST);
    getContentPane().add(buttonPanel, BorderLayout.SOUTH);
  }
} //end of PreferencesEditor
```

And that's how easy it is to implement a simple, yet usable, cross-platform registry editor. Already my mind is spinning with ideas on how to improve on this, like adding functionality to be able to modify the preference trees and making use of the Preferences API export/import capabilities (yhep, you can actually export preferences to XML files, and also import these files). A whole new preferable world is opening up...

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 031] - Hash, hash, away it goes!

Author: Dr. Heinz M. Kabutz
Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 31st issue of "The Java(tm) Specialists' Newsletter", where we look at how Hash Sets can become really confused. I love hearing from my readers, so please send your comments, praises, criticisms, accolades. The best compliment to me is if you forward this newsletter to other Java fans, for example to your local Java User Group (JUG).

I remember hearing last year that Visual Basic (shudder) was the fastest growing computer language. A report this year found that Java is now growing faster then Visual Basic, which frankly shudders me even more. There are apparently about 2 million people on this globe hacking away at our poor, dear language. Rather discouraging is that my newsletter is only reaching a meager 0.065% of Java programmers. Then again, this *is* supposed to be *The Java Specialists' Newsletter*, so maybe there aren't that many of us out there who should be considered specialists ;-)

How do you go from being an OO beginner to an OO guru? Simple answer: Experience! What if you can't wait 10 years to get that experience? Simple answer: Design Patterns! How can you learn Design Patterns? Simple answer: Ask me about my new course "Design Patterns - The Timeless Way of Coding".

Hash, hash, away it goes!

A few weeks ago, I received an email from *Charles N. May* who pointed out that it is quite easy to change an Integer using reflection in the same way that I showed in my article on Insane Strings. This is only possible because for some strange reason, or no reason at all, the data member value within the Integer class was not declared **final**.

Charles demonstrated the possible repercussions quite nicely by inserting the same Integer object several times into a java.util.HashSet, each time with different values. The contract of the java.util.Set says:

java.util.Set: *A collection that contains no duplicate elements. More formally, sets contain no pair of elements e1 and e2 such that e1.equals(e2), and at most one null element.*

It then goes on to say:

Note: Great care must be exercised if mutable objects are used as set elements. The behavior of a set is not specified if the value of an object is changed in a manner that affects equals comparisons while the object is an element in the set.

Interesting, but how do we define *mutable*, or *immutable* for that matter? I searched various Java texts but could not find a definition of what is meant by that. Let's try and define it for our use:

Immutable Class: A class where the state cannot be changed.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Simple enough, but what does *state* refer to? Is it a state descriptor of all the data members in the class? Or is it just the *identity* of the class? For example, let's consider the class Employee, which is part of the middle class ...

```
public class Employee {
  private final String name;
  private double salary;
  public Employee(String name, double salary) {
    this.name = name;
    this.salary = salary;
  }
  public String getName() { return name; }
  public double getSalary() { return salary; }
  public void setSalary(double salary) { this.salary = salary; }
}
```

Is the Employee class *mutable*? If we say that the state consists of a conglomeration of all the data members then the answer is "Yes". If we say the state consists of the identity, the answer is "No", as we cannot change the state. However, it will get even more confusing as we also have to define *identity*. I would say that the identity is the part of the class that determines the value of e1.equals(e2) and e1.hashCode(). So, in Employee above, what is the identity? Is it name? Nope, it is actually the location of the object in memory. If we want the identity to be according to the name, we will have to write equals() and hashCode() methods, the details of which are *below* the scope of TJSN.

Since *mutable* is so hard to pin down, we should actually rewrite the javadocs for the java.util.Set interface:

Note: Great care must be exercised if objects with a mutable identity *are used as set elements. The behaviour of a set is not specified if the value of an object is changed in a manner that affects equals comparisons* or its hashcode *while the object is an element in the set.*

By the description above, we should be careful with objects like String, Integer, etc. where the identity can be changed at runtime using reflection. Alright, it is bad coding practice to change private data members using reflection, but Integer could easily have fitted the above description if the value had been **final**. Then again, as discussed in my "final" newsletter, it is also bad coding practice to *not* mark data members final when they should be. Incidentally, java.lang.String is a lost cause - we can never make that properly immutable since its identity is contained in an array and arrays are mutable. Once the 1'998'700 monkeys who don't read this newsletter discover you can change Strings, we are all lost!

So, what happens when you change the identity of an object while it is in the HashSet? Let's have a look:

```
import java.lang.reflect.Field; // I can't resist using that package
import java.util.*;

public class HashHashGone {
  // Did you know that you can make a final static data member
  // without a value (i.e. blank), as long as you set the value
  // in the static initializer?  I love making every possible
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  // field final as it smokes out a lot of bugs.
  private static final Field intValue;
  static {
    try {
      intValue = Integer.class.getDeclaredField("value");
      intValue.setAccessible(true);
    } catch(NoSuchFieldException ex) {
      // we throw a RuntimeException from the static initializer
      // this will actually generate an Error and kill the thread
      // that first accessed GhostSet.
      throw new RuntimeException(
        "Serious error - no \"value\" field found in Integer");
    }
  }

  // This method changes the integer passed into the method to
  // the new value.
  private static void setInteger(Integer i, int newInt) {
    try {
      intValue.set(i, new Integer(newInt));
    } catch (IllegalAccessException ex) {
      throw new RuntimeException(
        "Serious error - field should have been accessible");
    }
  }

  // This method will be used later to print a detailed view of
  // the set, including the value, the class and the identity
  // hash code of the object.
  private static void printDetailed(Set set) {
    System.out.print("[");
    Iterator it = set.iterator();
    while(it.hasNext()) {
      Object val = it.next();
      System.out.print("\t(toString()=" + val +  ",class="
        + val.getClass().getName() +  ",identityHashCode="
        + System.identityHashCode(val) +  ")");
      if (it.hasNext()) System.out.print(", ");
      System.out.println();
    }
    System.out.println("]");
  }

  public static void main(String[] args) {
    if (args.length != 2) {
      System.out.println(
        "arguments: <initial-intvalue> <num-copies-to-insert-in-Set>");
      return;
    }
    int initialValue = Integer.parseInt(args[0]);
    int numberOfCopiesToInsert = Integer.parseInt(args[1]);

    Integer x = new Integer(initialValue);
    int currValue = initialValue;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    Set set = new HashSet();
    for (int i = 0; i < numberOfCopiesToInsert; i++) {
      setInteger(x, ++currValue);
      set.add(x);
    }
    setInteger(x, initialValue);
    System.out.println("here's a set containing " +
      numberOfCopiesToInsert + " copies of Integer(" + x + "): ");
    System.out.println(set);
    System.out.println("detailed view of set:");
    printDetailed(set);
    System.out.println("and does it contain that Integer?: " +
      set.contains(x));

    System.out.println("can the Integer be removed from the Set?");
    System.out.println(set.remove(x));
    System.out.println("the Set contents after attempted remove:");
    System.out.println(set);

    setInteger(x, -initialValue);
    System.out.println(
      "altering the Integer to its opposite makes the Set contain:");
    System.out.println(set);

    setInteger(x, initialValue);
    currValue = initialValue;
    for (int i = 0; i < numberOfCopiesToInsert; i++) {
      setInteger(x, ++currValue);
      set.remove(x);
    }
    System.out.println("now all the elements have been removed "
      + "from the Set as it contains:");
    System.out.println(set);
    System.out.println();
  }
}
```

When I run this with java HashHashGone 42 5, I get the following output:

```
here's a set containing 5 copies of Integer(42):
[42, 42, 42, 42, 42]
detailed view of set:
[       (toString()=42,class=java.lang.Integer,identityHashCode=6483656),
        (toString()=42,class=java.lang.Integer,identityHashCode=6483656),
        (toString()=42,class=java.lang.Integer,identityHashCode=6483656),
        (toString()=42,class=java.lang.Integer,identityHashCode=6483656),
        (toString()=42,class=java.lang.Integer,identityHashCode=6483656)
]
and does it contain that Integer?: false
can the Integer be removed from the Set?
false
the Set contents after attempted remove:
[42, 42, 42, 42, 42]
altering the Integer to its opposite makes the Set contain:
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

[-42, -42, -42, -42, -42]
now all the elements have been removed from the Set as it contains:
[]

The HashSet actually contains a HashMap that keeps the entries as keys and values. The HashMap contains a rehash() method, which is called when the table grows past the threshold determined by the number of entries and fill factor. However, contrary to what I believed until I tried it out today, the rehash method does not consider the possibility that there may be two objects in different hash positions, but equal, in the table. Re-hashing will therefore not solve the problem described above.

Moral of the story, assuming there is still morality left in this world? (btw, have you ever considered how similar the words mortal, moral and mutable are?) OK, the moral is? **Never** create a class where the identity can be changed once the object has been created.

That's all for this week. Please take a moment to think of how I could improve this newsletter and pop me an email.

Regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 032] - Exceptional Constructors - Resurrecting the dead

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 32nd issue of "The Java(tm) Specialists' Newsletter", the only Java newsletter for those of us who are considered *Java Specialists*. This week I want to look at what happens when an object is not properly initialised, the results are quite

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

interesting, I think. The ideas in this newsletter were spawned by Bruce Eckel, who sent me a piece of code that was throwing exceptions from the constructor and that made us both scratch our heads and rethink "what happens".

I must apologize for sending you two copies of the last two newsletters; a few of you pointed it out in a very nice manner (well-mannered bunch we are, hey?). When I sent out Issue030, I was toying with the idea of prefixing the subject line, so that you can easily write a junk mail filter for these newsletters. I started by having a prefix of [TJSN] but after I pressed the "Send" button, decided it would be better to use a prefix of [JavaSpecialists], so I pressed the IE stop button. I then changed the prefix and pressed send again, *voila!* two copies to everyone :-(. Issue031 was even worse, as my mailing list provider had changed the user interface, so as a result, the newsletter could not be read using my Outlook 2000! I changed what I thought was the problem and resent the newsletter together with an apology, only to discover that it still did not work. I do hope these problems are sorted out now.

The latest "skindernuus" (Afrikaans for gossip) is that Microsoft have now produced **J#** (J-sharp) which is a language using the Java syntax that can be compiled onto the .NET platform. This should make it really easy to port your Java programs onto a Microsoft system. Have a look at the DevX site. Is that bad news for me? Nope, it is fantastic news for me. Why? Because I have just finished writing a Design Patterns course on how to write Object-Oriented programs. I can now get customers who do Java, C++, C#, VB.NET *and* J# to come on my Design Patterns course. It's a classroom-based course and I will come and present it at your company anywhere in the world, if you can get together 10 programmers who wish to learn about Design Patterns from someone who's been "in the field" for a long time. Pop me an email if that interests you.

How do you go from being an OO beginner to an OO guru? Simple answer: Experience! What if you can't wait 10 years to get that experience? Simple answer: Design Patterns! How can you learn Design Patterns? Simple answer: Ask me about my new course "Design Patterns - The Timeless Way of Coding".

Exceptional Constructors

What happens when an Exception is thrown from within the constructor? Is the object created, or is it not created?

The answer is that the object is indeed constructed, but the handle to the object is not returned, so no-one will have a handle to it, so it can be garbage collected almost immediately. Naturally, when I hear something like "no-one", the wheels in my head start turning a bit faster.

What happens when you subclass a class that throws an exception out of its constructor? In the subclass, the first call has to be to the superclass, so if that throws an exception then the subclass cannot catch it.

Let's look at an example. We have a LicenseManager class that does some clever checking in its constructor and then throws an exception if the license is not valid.

```
public class LicenseManager {
  public LicenseManager() {
    if (!cleverLicenseValidation()) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      throw new SecurityException("License invalid");
    }
  }
  private boolean cleverLicenseValidation() {
    // here you would typically read the license
    // file, do some obfuscated calculations, etc.
    // and return true if the license is valid.
    return false;
  }
}
```

This class is used together with the SecuritySystem class to which we pass an instance of LicenseManager:

```
public class SecuritySystem {
  private static LicenseManager licenseManager;
  public static void register(LicenseManager lm) {
    // only register if it is not initialized
    if (licenseManager == null) {
      if (lm == null) {
        System.out.println("License Manager invalid!");
        System.exit(1);
      }
      licenseManager = lm;
    }
  }
}
```

The SecuritySystem would be initialized at some point with the LicenseManager and used in the Application

```
public class Application {
  public static void main(String[] args) {
    LicenseManager lm;
    try {
      lm = new LicenseManager();
    } catch(SecurityException ex) {
      lm = null;
    }
    SecuritySystem.register(lm);
    System.out.println("Now let's get things started");
  }
}
```

Yes, the code is rather convoluted, but I want to illustrate my point to you ;-). Let us assume, for the sake of this newsletter, that you cannot change any of the classes as they have been heavily obfuscated. Let's assume we want to attack this system by subclassing LicenseManager and having our own Application code load the Application. The problem we face is that if we try to subclass this LicenseManager with our own class, we cannot catch the exception from the parent. Let's have a look at the MyApplication class first:

```
public class MyApplication {
  public static void main(String[] args) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    MyLicenseManager lm;
    try {
      lm = new MyLicenseManager();
    } catch(SecurityException ex) {
      lm = null;
    }
    SecuritySystem.register(lm);
    // now we call the other application
    Application.main(args);
  }
}
```

Our first attempt at writing our own LicenseManager results in the following:

```
public class MyLicenseManager extends LicenseManager {
  public MyLicenseManager() {
    System.out.println("Created MyLicenseManager");
  }
}
```

When we run this, the result is:

License Manager invalid!

How do we get a handle to the MyLicenseManager class if it the parent constructor throws an exception?

An interesting use of finalize()

We could abuse finalize() to catch the handle to the MyLicenseManager class and then *resurrect* the object. (hihi). Now James Gosling says that you should never resurrect objects, but that is only because once resurrected, finalize will not be called again, which is actually what we want anyway. We therefore write a CleverLicenseManager that captures an instance in the finalize method:

```
public class CleverLicenseManager extends LicenseManager {
  private static CleverLicenseManager instance = null;
  public static CleverLicenseManager make() {
    try {
      new CleverLicenseManager();
    } catch(Exception ex) {} // ignore
    try {
      synchronized(CleverLicenseManager.class) {
        while (instance == null) {
          System.gc();
          CleverLicenseManager.class.wait(100);
        }
      }
    } catch(InterruptedException ex) {
      return null;
    }
    return instance;
  }
  public void finalize() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    System.out.println("In finalize of " + this);
    synchronized(CleverLicenseManager.class) {
      instance = this;
      CleverLicenseManager.class.notify();
    }
  }
  public CleverLicenseManager() {
    System.out.println("Created CleverLicenseManager");
  }
}
```

This is then created with the make() method from within our CleverApplication:

```
public class CleverApplication {
  public static void main(String[] args) {
    CleverLicenseManager lm = CleverLicenseManager.make();
    SecuritySystem.register(lm);
    // now we call the other application
    Application.main(args);
  }
}
```

The result is:

```
In finalize of CleverLicenseManager@77d134
Now let's get things started
```

Obviously, the CleverLicenseManager could have some data members that are not properly initialized, even final data members could be uninitialized.

Say the original LicenseManager class was marked as final. How do we make it non-final?

Making a class non-final without decompiling

In the *good ol' days* we were able to decompile any class with a simple decompiler such as JAD. More recently, Java programmers have become more nasty and have started using obfuscators that do terrible things to your class, making it a major effort to change a class from final to non-final.

Luckily the class format is well-known to us, so with a simple hex editor, we can make a class non-final without decompiling. Here's the dummy guide on how to do it:

Load the class in your favourite hex editor (e.g. frhed available at http://www.tu-darmstadt.de/~rkibria)
Look at the end of the method definitions, you will see a byte value that has bit 00010000 set, e.g. 0x31. Unset that bit and your class should now be non-final.

The non-dummy's guide would say that you should use a byte code library such as BCEL to change the class from final to non-final. Of course, if you're a freak like me, you'll just load the class in VI in binary mode and change the value of the correct character ;-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Disclaimer! = you shouldn't really hack licenses...

Please note that I do not under any circumstances advocate hacking license managers to deprive worthy code authors of their salaries. It is just an example, ok? And besides, no programmer would ever write a LicenseManager like the one I wrote ;-)

Cheersio

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 033] - Making Exceptions Unchecked

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 33rd issue of "The Java(tm) Specialists' Newsletter", which some cynics say should be renamed to "The Java(tm) Hackers' Newsletter". Yes, ok, that would probably be a more appropriate name considering some of our articles, but I just find that examining some of the weird behaviour in Java helps me to understand the language better.

Thanks to all of you who responded to my last survey where I asked you to send me email if you could *not* read my newsletter. You remind me of my good friend at university who won ALL the class medals for Computer Science. I once managed to beat him in a subject (parallel computing) where I got 90+% and he got 65%. He had thought he should answer 2 out of 4 questions, whereas he should have answered 3 out of 4 questions! The problem was that he did not *read the question* ;-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Meanwhile, my mailbox is quickly filling up with requests for information about the new Design Patterns course. If you have 10 or more programmers in your company that would like to attend such a course, please send me an email and I will see if I can pop around for a visit.

How do you go from being an OO beginner to an OO guru? Simple answer: Experience! What if you can't wait 10 years to get that experience? Simple answer: Design Patterns! How can you learn Design Patterns in a relaxed setting from someone who has used them in the real world? Simple answer: Ask me about my new course "Design Patterns - The Timeless Way of Coding".

Why Checked Exceptions Cause Bugs

How often have you seen code such as this?

```
// ...
try {
  // ... some code that could throw IOException or RemoteException
} catch(Exception ex) {} // ignore
```

Or even worse, how about the following (from a real program whose author shall remain anonymous):

```
Connection con;
try {
  // ... make a connection
} catch(Throwable t) {
  con = null;
}
```

You can imagine what happened. At some point, the piece of code that needed to make the connection tried to load a class that was not in the classpath, with a resulting NoClassDefFoundError. The program stopped working without informing us of what had gone wrong. After a while I noticed that the class could not have been included in the classpath and was horrified to find the above code.

Why is it that so many novice Java developers write code that catches Exception and then keeps quiet about it? Is it because the compiler *forces* us to catch them - and most beginners then don't know how to handle them correctly? I personally like checked exceptions because they tell me what things could go wrong in my environment, although not always very clearly. For example, java.io.IOException has 47 subclasses in JDK 1.3.1, so if a method throws IOException you really don't know what actually went wrong.

Sometimes I want to catch several checked exceptions and then do something about them as a group. The problem with catching Exception is that you then also catch the unchecked exceptions, i.e. derivatives of java.lang.RuntimeException, such as NullPointerException. I like to call the unchecked exceptions *programmer bug exceptions* and in general you would not want to swallow those exceptions. In that case I normally do the following:

```
// ...
try {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  // ... some code that could throw IOException or RemoteException
 } catch(RuntimeException ex) {
  // programmer bug - rethrow
  throw ex;
 } catch(Exception ex) {
  // handle all the exceptions correctly
 }
```

Incorrect handling of exceptions is the cause of most of the bugs I have seen in novice Java programmers' code. Seeing that most Java programmers are novices due to the age of the language, we could say that, even though I like them, checked exceptions could perhaps have been a mistake. C++ doesn't have them, neither does C# and also not Python. (In case you think I'm a great visionary for thinking that they might be a mistake, I got that idea from someone else ;-)

Wrapping Checked Exceptions in Unchecked Exceptions

So how do you convert a checked exception into an unchecked exception? Here we have a code sample of how one could write an **Object Adapter** that would convert a checked exception into a RuntimeException. (If you don't know the difference between an **object** and a **class** adapter, and what the **applicability** and the **consequences** of each are, you should seriously consider coming to my design patterns course - see beginning of newsletter for more information.)

One of the challenges of adapting exceptions is that you want to capture the exact place where the error occurred in the program, rather than the place where the checked exception was caught. The following would therefore *not* be the right way to do it, as you would have an incorrect line number to blame for the fault:

```
// incorrect way of adapting a checked exception
try {
  // ... some code that could throw IOException or RemoteException
 } catch(Exception ex) {
  throw new RuntimeException(ex.toString());
 }
```

After some experiments, I found that the ExceptionConverter shown below works fairly well.

```
/**
The ExceptionConverter changes a checked exception into an
unchecked exception.
*/
public class ExceptionConverter extends RuntimeException {
  /** we keep a handle to the wrapped exception */
  private final Exception ex;
  public ExceptionConverter(Exception ex) {
    this.ex = ex;
  }
  /** and allow the user of ExceptionConverter to get a handle to it. */
  public Exception getException() {
    return ex;
  }
  /** We print the message of the checked exception */
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public String getMessage() {
    return ex.getMessage();
  }
  /** and make sure we also produce a localized version */
  public String getLocalizedMessage() {
    return ex.getLocalizedMessage();
  }
  /** The toString() is changed to be prefixed with ExceptionConverter */
  public String toString() {
    return "ExceptionConverter: " + ex;
  }
  /** we have to override this as well */
  public void printStackTrace() {
    printStackTrace(System.err);
  }
  /** here we prefix, with s.print(), not s.println(), the stack
    trace with "ExceptionConverter:" */
  public void printStackTrace(java.io.PrintStream s) {
    synchronized (s) {
      s.print("ExceptionConverter: ");
      ex.printStackTrace(s);
    }
  }
  /** Again, we prefix the stack trace with "ExceptionConverter:" */
  public void printStackTrace(java.io.PrintWriter s) {
    synchronized (s) {
      s.print("ExceptionConverter: ");
      ex.printStackTrace(s);
    }
  }
  /** requests to fill in the stack trace we will have to ignore
  (I think)  We can't throw an exception here, as this method
  is called by the constructor of Throwable */
  public Throwable fillInStackTrace() {
    return this;
  }
}
```

We can try this out by throwing some exceptions around:

```
import java.io.IOException;
public class ExceptionConverterTest {
  private static void f() throws IOException {
    throw new IOException("File broken");
  }
  private static void g() {
    try {
      f();
    } catch(IOException ex) {
      System.out.println("Printing out plain ol' IOException:");
      System.out.println("---");
      ex.printStackTrace();
      System.out.println("---");
      throw new ExceptionConverter(ex);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
  }
  public static void main(String args[]) {
    try {
      g();
    } catch(RuntimeException ex) {
      System.out.println("Printing out RuntimeException:");
      System.out.println("---");
      ex.printStackTrace();
      System.out.println("---");
      System.out.println("That's it!");
    }
  }
}
```

The resulting output is:

```
Printing out plain ol' IOException:
---
java.io.IOException: File broken
        at ExceptionConverterTest.f(ExceptionConverterTest.java:5)
        at ExceptionConverterTest.g(ExceptionConverterTest.java:9)
        at ExceptionConverterTest.main(ExceptionConverterTest.java:20)
---
Printing out RuntimeException:
---
ExceptionConverter: java.io.IOException: File broken
        at ExceptionConverterTest.f(ExceptionConverterTest.java:5)
        at ExceptionConverterTest.g(ExceptionConverterTest.java:9)
        at ExceptionConverterTest.main(ExceptionConverterTest.java:20)
---
That's it!
```

As you can see, the line numbers for the two exceptions are the same, pointing us to the exact source of our troubles, without forcing us to use checked exceptions.

Is it a good idea to adapt checked exceptions into unchecked ones? In general I would say "no". However, there are some cases where you are overriding a method which does not throw a checked exception that you need to throw and in those cases it can sometimes be useful. How often have I morphed exceptions? Not often.

I'm looking forward to your comments on this newsletter.

Cheersio

Heinz

ERRATA

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

In my last newsletter I made a silly mistake - I forgot to set instance back to null. The make() method of the CleverLicenseManager class should have been:

```
public static CleverLicenseManager make() {
  synchronized(CleverLicenseManager.class) {
    try {
      new CleverLicenseManager();
    } catch(Exception ex) {} // ignore
    try {
      while (instance == null) {
        System.gc();
        CleverLicenseManager.class.wait(100);
      }
      return instance;
    } catch(InterruptedException ex) {
      return null;
    } finally {
      instance = null;
    }
  }
}
```

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 034] - Generic Types with Dynamic Decorators

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 34th issue of "The Java(tm) Specialists' Newsletter", in which we look at how we can apply the dynamic decorators to produce type-safe iterators dynamically. I know that some of you will be tempted to write to me how "that will all be in JDK 1.5" or "here are 1001 reasons why your construct is *not* useful". The

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

purpose of this newsletter is to make you think, please don't forget that. It is not to provide you with cut & dry solutions to your problems.

Thanks to Dr. Christoph Jung and Bruce Eckel for the ideas that spawned this newsletter. Some of these examples will probably be featured in the "Thinking in Patterns" book that Bruce is busy writing at the moment. I can't wait for *that* to get published ...

How do you go from being an OO beginner to an OO guru? Simple answer: Experience! But what if you can't wait 10 years to get that experience? Simple answer: Design Patterns! How can you learn Design Patterns in a relaxed setting from someone who has used them in the real world? Simple answer: Ask about my new course "Design Patterns - The Timeless Way of Coding".

1486 members are currently subscribed from 55 countries

Proxy vs. Decorator

What is the difference between a **Proxy** and a **Decorator**?

If we look at the structure of these patterns in the GoF book, we see that the Proxy has an association from the Proxy class to the RealSubject class (actually, it would be more versatile to have an association between Proxy and the Subject interface). Essentially the structure in Java code would look like this:

```
interface Subject {
  public void request();
}

class RealSubject implements Subject {
  public void request() { /* do something */ }
}

class Proxy implements Subject {
  private Subject realSubject;
  Proxy(Subject realSubject) {
    this.realSubject = realSubject;
  }
  public void request() {
    /* do something, then */
    realSubject.request();
  }
}
```

The Decorator pattern's structure has an aggregation from the Decorator class to the Component. The structure in Java code would look like this:

```
interface Component {
  public void operation();
}

class ConcreteComponent implements Component {
  public void operation() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    /* do something */
  }
}

class Decorator implements Subject {
  private Component component;
  Decorator(Component component) {
    this.component = component;
  }
  public void operation() {
    /* do something, then */
    component.operation();
  }
}

class ConcreteDecorator extends Decorator {
  ConcreteDecorator(Component component) {
    super(component);
  }
  public void anotherOperation() {
    /* decorate the other operation in some way, then call the
     other operation() */
    operation();
  }
}
```

If we now change the names of the classes to A, B, C and the method to f(), Proxy becomes:

```
interface A {
  public void f();
}

class B implements A {
  public void f() { /* do something */ }
}

class C implements A {
  private A a;
  C(A a) {
    this.a = a;
  }
  public void f() {
    /* do something, then */
    a.f();
  }
}
```

And if we do the same to Decorator, that becomes:

```
interface A {
  public void f();
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
class B implements A{
  public void f() { /* do something */ }
}

class C implements A {
  private A a;
  C(A a) {
    this.a = a;
  }
  public void f() {
    /* do something, then */
    a.f();
  }
}

class D extends C {
  D(A a) {
    super(a);
  }
  public void g() {
    /* decorate the other operation in some way, then call the
     other operation() */
    f();
  }
}
```

Oops, by looking at the structure alone, we see that there is not that much difference between Proxy and Decorator. We also have to look at the intents:

Proxy: Provide a surrogate or placeholder for another object to control access to it.

Decorator: Attach additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality.

We have to look at the structure and the intent and the applicability together if we want to figure out which pattern we are looking at. Even then, since a lot of the patterns are very similar, we can get into heated debates about this. Bruce Eckel and I exchanged some interesting emails trying to decide whether what you are about to see is the Decorator, Adapter or Proxy.

If you got lost in this discussion, you are missing out on one of the most powerful techniques developed in Software Engineering in the last decade, that being Design Patterns. I've developed a new classroom-based course that will teach you how to recognise and apply Object-Oriented Design Patterns. Please send me an email if you would like to know more.

The answer to the question of "what is the difference between a **Proxy** and a **Decorator**" lies mainly in the intent. We intend to use a Decorator to extend the interface of a object dynamically, whereas a Proxy is intended to provide a surrogate object for the real object.

Enough of a Design Patterns lesson. Let's get to the meat of today's newsletter.

Generic Types with Dynamic Decorators

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Bruce Eckel sent me an email asking if I had some nice examples for using the Dynamic Proxies of JDK 1.3. I passed the question on to my friend Christoph Jung who sent an interesting reply. He said that they use dynamic proxies quite a lot in order to handle type-safe collections. The issues addressed actually reminded me of generic collections, due in JDK 1.5 hopefully.

Let's remember how we used collections in the past. Say we had a stray animal pound in which we had collections of dogs and cats. This code is what the "client" code of our dogs and cats would use, i.e. in the client code we have to keep on doing type casting down to the correct type while iterating through the collection.

```
public class Cat {
  private final String species;
  public Cat(String species) {  this.species =  species; }
  public String toString() {  return species; }
  public void meow() {  System.out.println(this +  ": Meow"); }
}

public class Dog {
  private final String species;
  public Dog(String species) {  this.species =  species; }
  public String toString() {  return species; }
  public void bark() {  System.out.println(this +  ": Woof"); }
}

import java.util.*;
public class Pound {
  private Collection dogs;
  private Collection cats;
  public Pound(Dog[] dogs, Cat[] cats) {
    this.dogs =  Arrays.asList(dogs);
    this.cats =  Arrays.asList(cats);
  }
  public void makeNoise() {
    Iterator dog_it = dogs.iterator();
    while(dog_it.hasNext()) {
      ((Dog)dog_it.next()).bark(); // we have to downcast!
    }
    Iterator cat_it =  cats.iterator();
    while(cat_it.hasNext()) {
      ((Cat)cat_it.next()).meow(); // we have to downcast!
    }
  }
}
```

Wouldn't it be nice if we could have a collection for dogs and a collection for cats so that we would not have to worry about downcasting? Indeed, but Java does not support generics yet, I hear some of you say.

How could we improve our situation by abusing dynamic proxies? (For a detailed newsletter on dynamic proxies, please refer to http://www.smotricz.com/kabutz/Issue005.html) What if we decided on a naming convention within our objects that we want to collect? Let's say we change Dog to be the following:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public class Dog {
  private final String species;
  public Dog(String species) {  this.species =  species; }
  public String toString() {  return species; }
  public void bark() {  System.out.println(this +  ": Woof"); }
  public static interface Collection extends java.util.Collection {
    Iterator dogIterator();
  }
  public static interface Iterator extends java.util.Iterator {
    Dog nextDog();
  }
}
}
```

Our client code could therefore use Dog.Collection and call method dogIterator(), which would return a Dog.Iterator with the specialised method nextDog(). This means we never have to downcast the anonymous objects to Dog!

So, how do we actually make an instance of this Dog.Collection? Simple, dynamic proxies ... with a twist. First of all, I write a GenericFactory class (the Fowler Factory Method pattern, not the GoF). This GenericFactory has a makeCollection() method where I pass in an instance of the collection backing the Dog.Collection (this is where the adapter pattern comes in), and the type that I want to support. It is assumed that the type you pass in (e.g. Dog.class) has an inner interface called Collection with a method dogIterator() and another inner interface called Iterator that contains a method nextDog(). If these are not found we throw an IllegalArgumentException.

```
import java.lang.reflect.Proxy;
import java.util.*;
public class GenericFactory {
  public static Collection makeCollection(Collection backing,
      Class type) {
    GenericCollection gen =  new GenericCollection(backing, type);
    return (Collection)Proxy.newProxyInstance(
      gen.getTypeCollectionClass().getClassLoader(),
       new Class[] {  gen.getTypeCollectionClass() },
      gen);
  }
  /* please ignore makeIterator for now ... */
  public static Iterator makeIterator(Iterator backing, Class type) {
    GenericIterator gen =  new GenericIterator(backing, type);
    return (Iterator)Proxy.newProxyInstance(
      gen.getTypeIteratorClass().getClassLoader(),
       new Class[] {  gen.getTypeIteratorClass() },
      gen);
  }
}
```

The GenericCollection class looks something like this. What it does is intercept any calls to the Dog.Collection proxy and checks if it is the dogIterator() method. If it is, it uses the GenericFactory to make a Dog.Iterator.

```
import java.lang.reflect.*;
import java.util.*;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public class GenericCollection implements InvocationHandler {
  private final Collection backing;
  private final Class type;
  private final Class typeCollection;

  public GenericCollection(Collection backing, Class type) {
    this.backing = backing;
    this.type = type;
    typeCollection = discoverCollection();
  }

  /** Find the correct inner class Collection interface.
    @throws IllegalArgumentException if inner interface Collection
    not found */
  private Class discoverCollection() {
    Class[] innerClasses = type.getClasses();
    for (int i=0; i<innerClasses.length; i++) {
      if (innerClasses[i].getName().equals(
          type.getName() + "$Collection")) {
        return innerClasses[i];
      }
    }
    throw new IllegalArgumentException(
      "Class does not contain inner Collection interface");
  }

  public Class getTypeCollectionClass() {
    return typeCollection;
  }

  /** This is the meat of the GenericCollection.  It finds any
    method with type name followed by iterator and hijacks it. */
  public Object invoke(Object proxy, Method method, Object[] args)
      throws Throwable {
    if (method.getName().equalsIgnoreCase(type.getName() + "Iterator")) {
      return GenericFactory.makeIterator(backing.iterator(), type);
    }
    return method.invoke(backing, args);
  }
}
```

In order to make the picture complete, we also have to see the code that makes the Iterators. The GenericIterator looks very similar to the GenericCollection. I am sure we could have a GenericType from which both of those classes could inherit and which could contain common functionality. However, I was writing the code while trying to pacify a crying baby. Not that easy, seeing that I cleverly changed all the keys around on my notebook so that now the only way to use it is to touchtype. Almost dropped it a few times - the notebook, that is ;-)

```
import java.lang.reflect.*;
import java.util.*;

public class GenericIterator implements InvocationHandler {
  private final Iterator backing;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  private final Class type;
  private final Class typeIterator;

  public GenericIterator(Iterator backing, Class type) {
    this.backing = backing;
    this.type = type;
    typeIterator = discoverIterator();
  }

  /** Find the correct inner class Iterator interface.
    @throws IllegalArgumentException if inner interface Iterator
    not found */
  private Class discoverIterator() {
    Class[] innerClasses = type.getClasses();
    for (int i=0; i<innerClasses.length; i++) {
      if (innerClasses[i].getName().equals(
          type.getName() + "$Iterator")) {
        return innerClasses[i];
      }
    }
    throw new IllegalArgumentException(
      "Class does not contain inner Iterator interface");
  }

  public Class getTypeIteratorClass() {
    return typeIterator;
  }

  public Object invoke(Object proxy, Method method, Object[] args)
      throws Throwable {
    if (method.getName().equalsIgnoreCase("next" + type.getName())) {
      return backing.next();
    }
    return method.invoke(backing, args);
  }
}
```

How do you use this code? Let's have another look at the newly improved Pound class. You will notice that I am making the collections in the constructor of the Pound, and after that I use those collections directly. Normally, the collections would contain Value Objects from your database, and they would be made dynamically by your framework.

```
import java.util.*;
public class Pound {
  private Dog.Collection dogs;
  private Cat.Collection cats;
  public Pound(Dog[] dogs, Cat[] cats) {
    this.dogs = (Dog.Collection)GenericFactory.makeCollection(
      new LinkedList(Arrays.asList(dogs)), Dog.class);
    this.cats = (Cat.Collection)GenericFactory.makeCollection(
      new ArrayList(Arrays.asList(cats)), Cat.class);
  }
  public void makeNoise() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    Dog.Iterator dog_it = dogs.dogIterator();
    while(dog_it.hasNext()) {
     /* no more downcasting! */
     dog_it.nextDog().bark();
    }
    Cat.Iterator cat_it = cats.catIterator();
    while(cat_it.hasNext()) {
     /* no more downcasting! */
     cat_it.nextCat().meow();
    }
  }
}
}
```

We can test that in our PoundTest class by making a new Pound and calling the makeNoise method.

```
public class PoundTest {
  public static void main(String[] clargs) throws ClassNotFoundException {
    Pound spca =  new Pound(
      new Dog[] {
        new Dog("Alsation"),
        new Dog("Bulldog"),
        new Dog("PavementSpecial") },
      new Cat[] {
        new Cat("RussianBlue"),
        new Cat("DeadBounce") });
    spca.makeNoise();
  }
}
```

With output of:

```
Alsation: Woof
Bulldog: Woof
PavementSpecial: Woof
RussianBlue: Meow
DeadBounce: Meow
```

Oh, by the way, Cat would look like this:

```
public class Cat {
  private final String species;
  public Cat(String species) {  this.species = species; }
  public String toString() {  return species; }
  public void meow() {  System.out.println(this +  ": Meow"); }
  public static interface Collection extends java.util.Collection {
    Iterator catIterator();
  }
  public static interface Iterator extends java.util.Iterator {
    Cat nextCat();
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Yes, I know this is only a small step towards all the things you can do with generics in C++, and hopefully in JDK 1.5. However, it has made me think in a different way about what we can do with dynamic "proxies".

Was that a Proxy or a Decorator?

Back to our earlier discussion of whether this was a Proxy or a Decorator. I say this is more of a Decorator than anything else, as we are dynamically and transparently extending the functionality of existing classes without subtyping. The interesting part is that the Decorator class is not actually a class per se, it is a naming convention.

Disclaimer

I know that the dynamic decorators shown in this newsletters have some deficiencies at the moment. For example, it is possible to add a Cat to a collection of Dogs without an exception being thrown immediately. In a production-quality system you should obviously check that. **throw new** ExerciseForTheReaderException();

Last Words

Yes, I know that I am harping on about Design Patterns. However, without a solid understanding of Design Patterns you will never be a real Java Specialist. So, don't delay, pop me an email today ... :-)

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 035] - Doclets Find Bad Code

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 35th issue of "The Java(tm) Specialists' Newsletter". Two newsletters ago (http://www.smotricz.com/kabutz/Issue033.html) I mentioned

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

checked exceptions and that perhaps they were a mistake. I also mentioned that I got that idea from someone else, that someone actually having been Bruce Eckel. Bruce and I had some very lively and interesting discussions regarding checked exceptions prior to my newsletter, and some of the ideas were gleaned from those discussions. Bruce has put up a website on http://www.mindview.net/Etc/Discussions/CheckedExceptions, where you can now voice your opinion on that matter. Please join us and tell us what you think. On that forum I've posted a way in which you can switch off exception checking in the compiler *evil grin*.

I'm always grateful for good publicity for my newsletter, so thanks also to Chris Preimesberger of DevX for publishing the exceptions newsletter on their website (http://www.java-zone.com/free/articles/Kabutz03/Kabutz03-1.asp). Please remember to forward this newsletter to your friends and collegues.

How do you go from being an OO beginner to an OO guru? Simple answer: Experience! But what if you can't wait 10 years to get that experience? Simple answer: Design Patterns! How can you learn Design Patterns in a relaxed setting from someone who has used them in the real world? Simple answer: Ask about my new course "Design Patterns - The Timeless Way of Coding".

1618 members are currently subscribed from 55 countries

Doclets Find Bad Code

What makes code bad? Is it the result of mistakes in logic? That makes the code incorrect, but not bad. Bad is one of those fuzzy attributes that are hard to define. It makes your stomach turn, fills you with dread, confiscates your mind ...

What makes coders bad? Lack of experience? Sloppiness? Bad training at universities by lecturers who themselves are bad coders? I once made the mistake of suggesting that the company, at which I was working at the time, give a Dutch student an opportunity to get some real-life experience. The code was indescribably bad! All the data members used package access, every class was in the same package, the whole program was one long nested if-else statement.

One of the things that makes this hobby of newsletter writing satisfying is the feedback I get from readers. Some time in June this year I was busy working for a German company (Infor AG) developing a subcomponent based on Doclets (I will write more about that in another newsletter).

At the same time, one of my readers at CitiCorp in India sent me an email, asking me for help. She was working in the quality control department of the coding group and was looking for some easy way of enforcing company policies with regard to coding. Since I was working with Doclets anyway, we had some interesting discussions on how you could use Doclets to solve her problems.

On another continent, Bruce Eckel was busy writing solutions for the exercises in his book. Naturally, he also wanted to make sure that his exercises did not contain "bad code" by mistake, so this newsletter is the result of a cooperation between Java users across three continents :-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Let's get back to what makes code "bad". I believe that code is "bad" when it is difficult to maintain. In the refactoring literature, they say the code "smells". You might have code which is 100% correct, and yet still is "bad". How is that possible? If you don't believe me, wait until you have to change that code ...

There are certain "things" that we do which make code bad. Inappropriate variable names, making data members non-private, not having comments describing what each parameter is supposed to do, etc. But if you have a Java project with 2'000'000 lines of code, how do you find the code that is "bad"? Fortunately there is a solution that is not going to cost you the license fees of a Together/J, provided by Sun Microsystems in the form of Doclets.

The documentation for Doclets is in an obscure place. Go to your JDK documention, then to **Tool Docs**, click on **Basic Tools**, and under javadoc you'll see **Javadoc 1.3 Home Page**. All you have to do is write a method **public static boolean** start(RootDoc root) that is the starting point for the doclet engine. Let's have a look at some code that will find any non-private data members in your classes:

```
import com.sun.javadoc.*;

/** This class is used as a Doclet and will find any non-private
  data members in your classes. */
public class CodeChecker {
  /** This is the entry point for the doclet engine into your
  class. */
  public static boolean start(RootDoc root) {
    System.out.println("Non-private data members:");
    checkClasses(root.classes());
    return true;
  }
  /** We will call the checkClasses() method recursively so that
  we can also check the inner classes (for what it's worth). */
  private static void checkClasses(ClassDoc[] cds) {
    for (int i=0; i<cds.length; i++) {
      checkDataMembersArePrivate(cds[i]);
    }
  }
  /** This method prints out any data members that are not private
    together with their access.  If the field is package access
    we print out no modifiers. */
  private static void checkDataMembersArePrivate(ClassDoc cd) {
    System.out.println(cd.modifiers() + " " + cd.qualifiedName() + ":");
    FieldDoc[] fields = cd.fields();
    for (int i=0; i<fields.length; i++) {
      if (!fields[i].isPrivate()) {
        System.out.print("\t");
        if (!fields[i].isPackagePrivate())
          System.out.print(fields[i].modifiers() + " ");
        System.out.println(fields[i].type() + " " + fields[i].name());
      }
    }
    checkClasses(cd.innerClasses());
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The code above is dead-easy, but how do you use it? First you have to compile it, and you will need to include tools.jar in your classpath, e.g. javac -classpath .;c:\jdk1.3\lib\tools.jar *.java Note that the tools.jar file is not part of the runtime distributable, so if you want to use these features on your client machines, they will have to install the JDK, instead of just the JRE. Before we look at how we run Doclets, let's look at an example that we can use to test on:

```
public class Test {
  String packageAccess;
  private String privateAccess;
  public String publicAccess;
  protected String protectedAccess;
  public static final String NAME = "Hello";
}
```

We compile Test.java as well and now we can see how the Doclet is run:

```
javadoc -private -doclet CodeChecker Test.java
```

The resultant output is:

```
Loading source file Test.java...
Constructing Javadoc information...
Non-private data members:
public Test:
        java.lang.String packageAccess
        public java.lang.String publicAccess
        protected java.lang.String protectedAccess
        public static final java.lang.String NAME
```

You would have to of course match up your program with your company policy in order for this to be successful. For example, in your company you would probably say that it's OK for a data member to be **public** as long as it also was **static** and **final**.

More importantly than writing such a program is running it regularly, and suitably punishing those programmers found guilty of violating "the code" (in those countries that allow it, perhaps a **public** beheading would be appropriate? - oh no, rather a **private** one; otherwise they'd be violating their own rules.)

Would you use Doclets to do things to classes "in real time"? I would suggest you wait for the newsletter on how I tamed Doclets before you try that ;-)

My apologies again for not getting this newsletter out as regularly as I would like to. My excuse is rather lame - I'm busy writing a program for a company, and I love programming so much that everything else takes second place. The coding was also quite a rushed job, which is why I'm busy on a Saturday afternoon writing a newsletter when all my colleagues are on the beach ;-)

On the 30th of November we are celebrating our 1st anniversary as "The Java(tm) Specialists' Newsletter". I would really appreciate it is you could make an extra effort to spread the word before that, so we can see some good growth.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Kind regards, and until next time...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 036] - Using Unicode Variable Names

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 36th edition of "The Java(tm) Specialists' Newsletter". This week, we will look at the strange things that happen when we try to use unicode characters in our code.

I am sitting outside in my garden, with beautiful sunshine and a pitbull terrier at my command ;-) Approximately a month ago, the biggest software vendor in South Africa went bankrupt, severely affecting the availability of software in this country. Fortunately for me, I have friends in convenient places: I purchased the software that I needed (Dragon NaturallySpeaking) from Amazon in Germany and had it shipped to infor AG, who I have spoken about in other newsletters - they very kindly shipped it down to the end of the earth.

As a result of using Dragon NaturallySpeaking, you will probably notice that my newsletters will have an even more conversational style than before. I am always looking at ways in which I can improve my newsletters and serve you better. Please remember to forward this newsletter to friends and colleagues who are interested in Java.

A special welcome to country No 56, Malta! My wife's previous boss at a hotel was the Maltese ambassador for Cape Town, which was really cool, as he had diplomatic

immunity from parking fines and speeding fines. Mind you, traffic laws are rather lax in this country, I have only had one speeding fine in my life, and I drive an Alfa Romeo!

South Africa has just become the cheapest country in the world! We are the first country where a Big Mac costs less than US$ 1. It is cheaper here even than in the Philipines and China. I had a good response to my advert for my Java Course (thank you for your patience in this regard) and so I definitely want to develop the idea of running courses in South Africa, combined with a holiday :-)

How do you go from being an OO beginner to an OO guru? Simple answer: Experience! But what if you can't wait 10 years to get that experience? Simple answer: Design Patterns! How can you learn Design Patterns in a relaxed setting from someone who has used them in the real world? Simple answer: Ask about my new course "Design Patterns - The Timeless Way of Coding".

1707 members are currently subscribed from 56 countries

Using Unicode Variable Names

A few months ago, I was reading a book written by the authors of Java, when I stumbled across a piece of code that was using Unicode characters as variable names. Being the curious type, I immediately tried writing a piece of code that used funny characters. Easier said than done! I don't know of any Java IDE that supports Unicode. The common e-mail systems in this world would also choke like a dog on a chicken bone if I sent you a newsletter containing Unicode characters ;-)

Before I get into *how* we could use Unicode characters in our variables, let's just take a step back and think about it: Imagine being called in by a Japanese company who has got a memory leak in their program which they want you to fix (one of the most common tasks I have been asked to perform), and imagine if in their company they used Japanese characters for their variables. Yes, it would compile if you follow the ideas in this newsletter, but what would the result be for me? I would probably pack my bags and head back home! It's bad enough having to read code where the variable names are in German or in Afrikaans, I cannot imagine trying to understand code where I don't even know the characters used in variable names!

Since I could not find an IDE that supported Unicode, my first job was to write a Unicode editor. Also easier said than done. I had learned many years ago that Writers and Readers are used for Unicode characters, but I had never really used Unicode before. My first approach at reading and writing Unicode files looked something like this:

```
public void load() throws IOException {
  BufferedReader in = new BufferedReader(new FileReader(filename));
  String s;
  while((s = in.readLine()) != null) {
    // ...
  }
}
```

Did you know that FileReader extends InputStreamReader? In its constructor it constructs a FileInputStream that it passes to its parent. The InputStreamReader has

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

a constructor that takes as argument the encoding used for reading files. FileReader unfortunately does not expose the constructor that takes the encoding as an argument, it simply uses an operating-system dependent encoding. One cannot but wonder what the author of the FileReader had been smoking the day he/she wrote that code ...

(Actually, when I wrote the Sun Microsystems Java programmer examination a few years ago, the only none-GUI question that I got wrong was a question relating to reading ISO-8859-1 data. Perhaps there has always been a hole in my knowledge regarding this topic.)

Should you want to use the FileReader to read an encoding different to the standard one, you would have to do the following:

```
public void load() throws IOException {
  BufferedReader in = new BufferedReader(
    new InputStreamReader(
      new FileInputStream(filename), "UTF-16BE"));
  String s;
  while((s = in.readLine()) != null) {
    // ...
  }
}
```

Without further ado, here is the code for a Unicode text editor. It allows you to insert Unicode characters by entering their decimal values and pressing the appropriate button. For the design, I have followed an approach I saw a few years ago on jGuru, where all the GUI elements are created lazily. It makes the GUI code very nicely maintainable, as you never have to worry in what order elements are constructed.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class UnicodeEditor extends JFrame {
  private JPanel buttonPanel;
  private JScrollPane editorPanel;
  private JTextArea editor;
  private final String filename;
  private final String encoding;

  public UnicodeEditor(String filename, String encoding)
      throws IOException {
    this.filename = filename;
    this.encoding = encoding;
    getContentPane().add(getButtonPanel(), BorderLayout.NORTH);
    getContentPane().add(getEditorPanel(), BorderLayout.CENTER);
    load();
  }

  protected JPanel getButtonPanel() {
    if (buttonPanel == null) {
      buttonPanel = new JPanel();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    JButton unicodeInsert = new JButton("Insert Unicode:");
    final JTextField unicodeField = new JTextField(8);
    JButton saveExit = new JButton("Save & Exit");
    unicodeInsert.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        getEditor().insert(
          "" + (char)Integer.parseInt(unicodeField.getText()),
          getEditor().getCaretPosition());
      }
    });
    saveExit.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        try {
          save();
          System.exit(0);
        } catch(IOException ex) { ex.printStackTrace(); }
      }
    });
    buttonPanel.add(unicodeInsert);
    buttonPanel.add(unicodeField);
    buttonPanel.add(saveExit);
  }
  return buttonPanel;
}

protected JTextArea getEditor() {
  if (editor == null) {
    editor = new JTextArea();
  }
  return editor;
}

protected JScrollPane getEditorPanel() {
  if (editorPanel == null) {
    editorPanel = new JScrollPane(getEditor());
  }
  return editorPanel;
}

protected void load() throws IOException {
  BufferedReader in = new BufferedReader(new InputStreamReader(
    new FileInputStream(filename), encoding));
  StringBuffer buf = new StringBuffer();
  int i;
  while((i = in.read()) != -1) buf.append((char)i);
  in.close();
  getEditor().setText(buf.toString());
}

protected void save() throws IOException {
  BufferedWriter out = new BufferedWriter(new OutputStreamWriter(
    new FileOutputStream(filename), encoding));
  char[] text = getEditor().getText().toCharArray();
  for (int i=0; i<text.length; i++) out.write(text[i]);
  out.close();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }

  public static void main(String[] args) throws IOException {
    if (args.length < 1)
      throw new IllegalArgumentException(
        "usage: UnicodeEditor filename [encoding]");
    String encoding = (args.length == 2)?args[1]:"UTF-16BE";
      UnicodeEditor editor = new UnicodeEditor(args[0], encoding);
    editor.setSize(500,500);
    editor.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    editor.show();
  }
}
```

By default this uses the UTF-16BE format, standing for *Sixteen-bit Unicode Transformation Format, big-endian byte order*. You can specify any encoding when you start the editor, such as UTF-8, ISO-8859-1, etc. But, before we use this editor, we first need to have a file containing Unicode characters. I've written a code generator that generates two files, MathsSymbols.java and MathsSymbolsTest.java:

```
import java.io.*;
public class UnicodeVariableGenerator {
  public static void generateMathsSymbols() throws IOException {
    PrintWriter out = new PrintWriter(new OutputStreamWriter(
      new FileOutputStream("MathsSymbols.java"), "UTF-16BE"));
    out.println("public interface MathsSymbols {");
    out.print("  public static final double ");
    out.print((char)960);
    out.println(" = 3.14159265358979323846;");
    out.print("  public static final double ");
    out.print((char)949);
    out.println(" = 2.7182818284590452354;");
    out.println("}");
    out.close();
  }
  public static void generateMathsSymbolsTest() throws IOException {
    PrintWriter out = new PrintWriter(new OutputStreamWriter(
      new FileOutputStream("MathsSymbolsTest.java"), "UTF-16BE"));
    out.println("public class MathsSymbolsTest implements MathsSymbols {");
    out.println("  public static void main(String args[]) {");
    out.println("    System.out.println(\"The value of PI is: \" + \u03C0);");
    out.println("    System.out.println(\"The value of E is: \" + \u03B5);");
    out.println("  }");
    out.println("}");
    out.close();
  }
  public static void main(String[] args) throws IOException {
    generateMathsSymbols();
    generateMathsSymbolsTest();
  }
}
```

I won't include the code for MathsSymbols.java and MathsSymbolsTest.java, please run the UnicodeVariableGenerator class to generate that code. I already bomb out

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

enough mailing systems by sending my newsletters in HTML (*evil grin*), no use in causing more trouble by using Unicode. Once you've run the UnicodeVariableGenerator, please load the MathsSymbols.java file with the UnicodeEditor, using UTF-16BE and have a look at it: you should see the Greek symbol for PI.

The last "trick" you need to know about is how to compile the MathsSymbols.java and MathsSymbolsTest.java. If you open the files with notepad or vi, you will probably see a rather strangely formatted file, with two bytes being used per character. When you compile these files, you therefore have to specify the character encoding used:

javac -encoding UTF-16BE MathsSymbols*.java

That's it! And it has kept me busy longer than just about all the other newsletters to try and get it right. Another interesting variation of this is where David Treves (who I met through a really cool advanced Java chat list - JavaDesk on YahooGroups - where you get shouted at if you ask beginner questions) tried to write/read Hebrew to the Database. He doggedly tried to get it working until eventually he succeeded - after I had given up hope of ever figuring it out. Stay tuned for the next few weeks to see how he did it.

Until next week, when we celebrate our first anniversary as the most interesting Java newsletter on the Internet ;-)

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 037] - Checking that your classpath is valid

Author: Sydney Redelinghuys (with Dr. Heinz M. Kabutz in the peanut gallery)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 37th edition of "The Java(tm) Specialists' Newsletter", sent to 2103 Java experts in 62 countries. This week, my friend Sydney Redelinghuys has stepped in and provided the ideas for this newsletter, as well as most of the writing. Sydney was the brain behind the SoftHashMap newsletter that also made it onto DevX. I consider Sydney to be the 2nd best Java programmer that I know (position #1 is already given to Paul), [SR: So what position is given to HK] [HK: Position # 0 of course! ;-] but I marvel at the designs Sydney comes up with, I don't know anyone with such a knack for cool designs. Sydney found a copy of **the** Design Patterns book in the home economics section of his university (I will refrain from making comments regarding his university based on that last comment ;-))).

How do *you* go from being an OO beginner to an OO guru? Simple answer: Experience! But what if you can't wait 10 years to get that experience? Simple answer: Design Patterns! How can you learn Design Patterns from someone who has used them in the real world? Simple answer: Ask about my new course "Design Patterns - The Timeless Way of Coding".

Enough of my jabbering, let's see what Sydney, the one-legged Java Guru has to say ...

Checking Your Classpath Validity

The classpath - like most programmers, I really struggled with it in the beginning. I first cut my teeth on Java in middle of 1996 when I was a student doing some vacation work, as part of an experiment of using JDK 1.0 to solve the world's problems. As you might imagine, I caused more problems than I solved. I must've been the most underpaid Java programmer of all time - in today's South African currency terms, I only got US$ 272 for 2 months work (and that was with a 100% bonus)! Enough reminiscing, back to class paths. It is not that class paths are difficult to understand, it is just that it is too easy to get them wrong and not easy enough to locate errors, especially if you are used to case-insensitive NT paths. The classpath problem is so part of the Java culture that even The Onion reports on it! [HK: A must-read magazine for TJSN subscribers]

Even after I got more experienced it still seemed to eek its way into builds. In those days (before Ant and what not) we used to write huge batch files to check that all the required jars were available, I believe the company in question still does it ;) [HK: Actually, we had a chap who had an MCSE in WinNT 3.51 who was a real whizz at batch files. This, perhaps, was to our disadvantage.]

These days I am involved with Websphere, a product aimed at the Java professional who thinks he knows how Java works. This product will make even the seasoned Java expert quiver at his knees when he gets a MarshallException at run-time due to linkage errors often caused by an incorrect classpath. Note: in Websphere the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

classpath can be specified on at least **three** levels (container, server and node levels).

My first attempt at solving the marshall exception was to add another dynamic layer like any true geek developer. [HK: Sydney is truly the master of adding dynamic layers. That first Java program that he wrote for US$ 272 over 2 months took US$ 13500 of *real professional* Java programmers' time to debug.] Java does tend to make it too easy to do this with reflection, dynamic class loaders, dynamic proxies, soft references and all other kinds of weird & wonderful constructs this newsletter likes to comment on.

So, my first solution was to load the class files dynamically (via Class.forName(...)) and then to simply try catching the ClassNotFoundException.

```
try {
  Class my_class = Class.forName("ClassName");
} catch(ClassNotFoundException cnfe) {
  cnfe.printStackTrace();
}
```

Of course this worked, so now I could see the exception other application servers probably would have provided me with originally.

This really bugged me. I had to write compilers at varsity and I always thought I was doing the world a favour by doing compile time type checking etc. It surely is better to find a bug at compile time than at run-time, especially if your build-deploy cycle spirals into an affair that can take hours. [HK: With the batch files that we had originally the build cycle took so long that we could only do a weekly build!] [SR: I think the most time consuming part of that process was actually Source Safe]

Ant (or other similar build-tools) could solve some of these problems, unfortunately some of our components have to be deployed manually due to our IDE keeping information hidden. [HK: Isn't your IDE written in Java - perhaps you could use reflection to find the information dynamically and ...] [SR: No such luck it is actually written in SmallTalk.]

So I asked myself: *"Why does Sun keep the classpath hidden at runtime?"* I decided to start digging in the jdk api's.

To load classes, the default class loader needs access to the classpath. The default class loader is very easy to get hold of, via the java.lang.ClassLoader.getSystemClassLoader static method. Unfortunately, the interface of java.lang.ClassLoader does not contain any methods that appear to be useful to us.

Just before I decided to start digging in the Sun source code, I noticed that java.security.SecureClassLoader extended java.lang.ClassLoader and java.net.URLClassLoader extended java.security.SecureClassLoader. By doing a quick experiment, I discovered that the System class loader (in the SUN VM) is indeed an instance of java.net.URLClassLoader. [HK: *evil grin*]

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

At this point I got all excited. Why? Because URLClassLoader contains a method getURLs() which returns the current classpath. So by writing the following small class, I can validate the classpath (i.e. make sure all the jars exist).

```
import java.net.*;
import java.io.*;

public class ClassPathInfo {
  private final ClassLoader classLoader;

  public ClassPathInfo(ClassLoader classLoader) {
    this.classLoader = classLoader;
  }

  public ClassPathInfo() {
    this(ClassLoader.getSystemClassLoader());
  }

  /**
   * validates classpath, throws an LinkageError if invalid
   * classpath was specified
   */
  public void validateClassPath() {
    try {
      URL[] urls = ((URLClassLoader)classLoader).getURLs();
      for(int i=0; i<urls.length; i++) {
        try {
          urls[i].openStream();
        } catch(IllegalArgumentException iae) {
          throw new LinkageError(
            "malformed class path url:\n "+urls[i]);
        } catch(IOException ioe) {
          throw new LinkageError(
            "invalid class path url:\n "+urls[i]);
        }
      }
    } catch(ClassCastException e) {
      throw new IllegalArgumentException(
        "The current VM's System classloader is not a "
        + "subclass of java.net.URLClassLoader");
    }
  }

  public static void main(String[] args) {
    ClassPathInfo info = new ClassPathInfo();
    info.validateClassPath();
  }
}
```

Now it is possible to add batch or script files to replace java and javac. I was planning to leave this as an exercise for the reader but Heinz would have none of that [HK: Incidentally, someone sent me the "exercise to the reader" parts of the CircularArrayList, if any of you are interested]. It turned out to be more difficult than it initially seemed, as it is tricky to make sure that ClassPathInfo is in the classpath.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Firstly if you are a Unix user I shall leave this as an exercise for the reader. [HK: awww, come-on Syd] If you are using Windows you need to copy the following three batch files with ClassPathInfo.class to somewhere on your PATH. You also need to change the STRICTHOME enviroment variable in checkcp.bat to the directory these files were copied into.

checkcp.bat

```
@echo off
rem Set STRICTHOME to the directory you're deploying this file to.
set STRICTHOME=C:\strictjava

if ""=="%3" (
  set STRICTCP="%CLASSPATH%;%STRICTHOME%"
) else (
  if -classpath==%1 (
    set STRICTCP="%2;%STRICTHOME%"
  )else (
    echo If more than three parameters are specified,
    echo the first has to be -classpath.
    set ERRORLEVEL=1
    goto end
  )
)

java -classpath %STRICTCP% ClassPathInfo
set STRICTHOME=
set STRICTCP=

:end
```

strictjavac.bat

```
@echo off
call checkcp %1 %2 %3
if not %errorlevel%==0 goto END
javac %1 %2 %3 %4 %5 %6 %7 %8 %9
:END
```

strictjava.bat

```
@echo off
call checkcp %1 %2 %3
if not %errorlevel%==0 goto END
java %1 %2 %3 %4 %5 %6 %7 %8 %9
:END
```

Please note that if you supply the classpath to these batch files, you will have to put it in quotes (i.e. strictjava -classpath "c:\sub\jar1.jar;c:\sub2\jar2.jar" MyClass).

Regards

Sydney Redelinghuys (aka hopalong)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 038a] - Counting Objects Clandestinely

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 38th edition of "The Java(tm) Specialists' Newsletter", sent to 2226 Java experts in 70 countries, with Egypt as latest addition. I'm missing a few African countries, ok, it's worse than ***missing***, I only have South Africa, Ghana, Nigeria, Zimbabwe and Egypt, so if you are from another African country, please send me an email.

Talking about Africa, I didn't include Mauritius in my list of African countries, although politically I probably should have. If you don't know what Mauritius is, it is a small island in the Indian Ocean inbetween South Africa and India. Mauritius was colonised by the French (who only drank and fought and never did much for the country except introduce bureaucracy ;-) and later properly colonised by the British. The result is that English is the official language, whereas French (and derivatives thereof) is the spoken language.

The people in Mauritius are amazing. Extremely friendly, and not only to sell you things. The beaches are fantastic, the water warm, lots of things to see and do, such as waterskiing, etc. Oh, and the hotels! Such service you've never seen anywhere. So, surprise, surprise, I am running my course "Design Patterns - The Timeless Way of Coding" on that beautiful island in February through a Mauritian company (yes, life is tough for us Java programmers ;-).

Counting Objects Clandestinely

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

A few months ago I received a frantic phone call from a friend who had gotten himself seriously stuck. His program was running out of memory and he didn't know where the objects were being created. He ran his program in a commercial memory analyser, which only had the effect that the memory analyser crashed. (If you want to have a laugh, try running any big Java application (e.g. Together/J or JBuilder 4) in a memory analyser). *If* it runs, you'll see that the success of your product does not depend on good programming but on good marketing.

The trick I'm going to show you in this newsletter essentially saved my friend's project, and it is sooo easy and cheap, you won't believe it. Please note however, that this is only an initial implementation and that you should extend the functionality if you want to use it a lot. I have myself used this idea to effectively find memory problems in my own applications.

Imagine for a second that there was some way of knowing when an Object was created. Well, if you really want that functionality, why don't you add it? Yes, why don't you change java/lang/Object? Sure, it's intrusive, but very effective. The test code that you add to your code to find the memory leaks would be removed anyway (or commented out?) before the final build, so why not? It took me a while to get all this right, which is why you didn't see a newsletter for a while, but here is the code, which I tried on Windows JDK 1.3.1. On other VMs it might reformat your harddrive, sell your credit card details or promote you to sales manager.

```
package java.lang;
import java.util.*;
public class Object {
 // ... stick this at the bottom of Object.java:
 private static HashMap countMap = new HashMap();
 private static boolean counting = true;
 private static int totalObjects = 0;

 private class Counter {
  int value = 1;
 }

 public Object() {
  synchronized(Object.class) {
   if (counting) {
    counting = false;
    totalObjects++;
    Counter c = (Counter)countMap.get(getClass());
    if (c == null) {
     countMap.put(getClass(), new Counter());
    } else {
     c.value++;
    }
    counting = true;
   }
  }
 }
 public synchronized static void ___resetObjectCreationStats() {
  counting = false;
  totalObjects = 0;
  countMap.clear();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    counting = true;
  }
  public static void ___printObjectCreationStats() {
    ___printObjectCreationStats(System.out);
  }
  public synchronized static void ___printObjectCreationStats(java.io.PrintStream
out) {
    counting = false;
    out.println("Total number of objects: " + totalObjects);
    TreeSet sorted = new TreeSet(new Comparator() {
      public int compare(Object o1, Object o2) {
        int value1 = ((Counter)((Map.Entry)o1).getValue()).value;
        int value2 = ((Counter)((Map.Entry)o2).getValue()).value;
        int result = value2 - value1;
        if (result == 0) {
          String classname1 = ((Class)((Map.Entry)o1).getKey()).getName();
          String classname2 = ((Class)((Map.Entry)o2).getKey()).getName();
          return classname1.compareTo(classname2);
        }
        return result;
      }
    });
    sorted.addAll(countMap.entrySet());
    Iterator it = sorted.iterator();
    while(it.hasNext()) {
      Map.Entry entry = (Map.Entry)it.next();
      out.println("\t" + ((Counter)entry.getValue()).value
        + "\t" + ((Class)entry.getKey()).getName());
    }
    out.println();
    counting = true;
  }
}
```

You can now know exactly how many objects were created since you last reset the counters. [Actually, not really exactly. This approach will not tell you when arrays are created, but you will find out about any non-array objects that were made.] The way you would use this in your code is as follows:

```
import java.util.*;
public class Test {
  public static void testStringConcatenation() {
    ___resetObjectCreationStats();
    int localHost = 0x7F000001;
    String s =
      ((localHost>>24)&0xff) + "."
      + ((localHost>>16)&0xff) + "."
      + ((localHost>>8)&0xff) + "."
      + (localHost&0xff);
    System.out.println("Objects created to make an IP address String");
    System.out.println("----------------------------------------");
    ___printObjectCreationStats();
  }
  public static void testLinkedListCreation() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    ___resetObjectCreationStats();
    LinkedList daysOfWeek = new LinkedList();
    System.out.println("Objects created to make a LinkedList");
    System.out.println("-----------------------------------");
    ___printObjectCreationStats();
    daysOfWeek.add("Sunday");
    daysOfWeek.add("Monday");
    daysOfWeek.add("Tuesday");
    daysOfWeek.add("Wednesday");
    daysOfWeek.add("Thursday");
    daysOfWeek.add("Friday");
    daysOfWeek.add("Saturday");
    System.out.println("and after adding 7 elements to the LinkedList");
    System.out.println("-------------------------------------------");
    ___printObjectCreationStats();
  }
  public static void testHashMapCreation() {
    ___resetObjectCreationStats();
    HashMap monthsVsLength = new HashMap();
    System.out.println("Objects created to make a HashMap");
    System.out.println("--------------------------------");
    ___printObjectCreationStats();
    monthsVsLength.put("January", new Integer(31));
    monthsVsLength.put("February", new Integer(28));
    monthsVsLength.put("March", new Integer(31));
    monthsVsLength.put("April", new Integer(30));
    monthsVsLength.put("May", new Integer(31));
    monthsVsLength.put("June", new Integer(30));
    monthsVsLength.put("July", new Integer(31));
    monthsVsLength.put("August", new Integer(31));
    monthsVsLength.put("September", new Integer(30));
    monthsVsLength.put("October", new Integer(31));
    monthsVsLength.put("November", new Integer(30));
    monthsVsLength.put("December", new Integer(31));
    System.out.println("and after adding 12 elements to the HashMap");
    System.out.println("-------------------------------------------");
    ___printObjectCreationStats();
  }
  public static void main(String args[]) {
    System.out.println("Objects created to get the VM started");
    System.out.println("-----------------------------------");
    ___printObjectCreationStats();
    testStringConcatenation();
    testLinkedListCreation();
    testHashMapCreation();
  }
}
```

When you compile the test class you should also at the same time point it to your Object.class otherwise you'll get a compiler moan. You have to run Test by including the new Object.class file in your bootclasspath, for example java -Xbootclasspath/p:. Test, which will prepend the current directory to the boot classpath. The output is rather long and would be different for each VM version.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
Objects created to get the VM started
-------------------------------------
Total number of objects: 784
      514    java.util.Hashtable$Entry
      104    java.lang.String
       25    java.lang.StringBuffer
       22    java.util.Locale
       13    java.io.File
       11    sun.security.action.GetPropertyAction
* snip - run it yourself to see the rest*

Objects created to make an IP address String
--------------------------------------------
Total number of objects: 6
       5    java.lang.String
       1    java.lang.StringBuffer

Objects created to make a LinkedList
------------------------------------
Total number of objects: 2
       1    java.util.LinkedList
       1    java.util.LinkedList$Entry

and after adding 7 elements to the LinkedList
---------------------------------------------
Total number of objects: 9
       8    java.util.LinkedList$Entry
       1    java.util.LinkedList

Objects created to make a HashMap
---------------------------------
Total number of objects: 1
       1    java.util.HashMap

and after adding 12 elements to the HashMap
-------------------------------------------
Total number of objects: 25
      12    java.lang.Integer
      12    java.util.HashMap$Entry
       1    java.util.HashMap
```

What are the disadvantages of this approach, and what are the alternatives? The disadvantages that I can think of immediately are that you don't see the arrays like this and that you cannot see where objects were created from. I don't know how to solve the problems with the arrays, but you could easily change the code to also remember the stack trace of each object that gets constructed. You could then analyse the stack traces and find out exactly where the objects are constructed from. The alternatives I can think of is to use java -Xrunhprof (to perform JVMPI heap, cpu, or monitor profiling) or to use an expensive commercial tool. The disadvantage of some of the commercial tools is that if your program quickly eats memory the memory tools tend to fall on their back and play dead.

Have you ever wondered why Swing is so slow? Have a look at this test code:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
import javax.swing.*;
import java.awt.BorderLayout;
import java.awt.event.*;

public class TestGUI extends JFrame {
  public TestGUI() {
    super("TestGUI");
    getContentPane().add(new JTextArea(), BorderLayout.CENTER);
    JButton press = new JButton("Press Me!");
    press.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        ___printObjectCreationStats();
        ___resetObjectCreationStats();
      }
    });
    getContentPane().add(press, BorderLayout.SOUTH);
    setSize(500, 600);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    show();
  }
  public static void main(String[] args) {
    new TestGUI();
  }
}
```

I ran this code and moved my mouse in circles for 10 seconds in the JTextArea. Here are the first couple of entries that were shown when I pressed the "Press Me!" button:

```
Total number of objects: 15622
   2867   java.lang.String
   1545   java.lang.ref.Finalizer
   1384   java.awt.event.MouseEvent
   1326   java.awt.Point
   1200   java.lang.StringBuffer
   1047   java.util.Hashtable$Entry
   769    java.util.WeakHashMap$WeakKey
   750    java.awt.EventQueueItem
   706    sun.awt.EventQueueItem
   648    java.awt.Rectangle
   316    sun.java2d.loops.GraphicsPrimitiveProxy
   266    sun.awt.MostRecentKeyValue
   253    java.awt.geom.AffineTransform
   134    java.util.HashMap$Entry
```

Why so many Strings? I can only assume that that has to do with the pluggable look & feel. The reason I say that is because if I press the button again (after moving the mouse in circles a few times), I get the following output:

```
Total number of objects: 5257
   891    java.lang.ref.Finalizer
   831    java.awt.event.MouseEvent
   796    java.awt.Point
   456    java.util.WeakHashMap$WeakKey
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

436 java.awt.EventQueueItem

It gets quite interesting when we look at different VMs, just remember the warning at the beginning of this newsletter ;-)

That's all for this week and this year. I wish you all the best for 2002, may you find much opportunity to tell others about this newsletter ;-) [and make me prosper in the process ;]

Kind regards, and now I must get back to spending a few sunny days with my wife & two kiddies.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 038b] - Counting Objects Clandestinely - Follow-up

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

When I wrote the last newsletter, I still had too much sea water in my head with the effect that I actually didn't write as much in terms of explanation as I should have. Consequently, some of you sent me questions that made me realise that a follow-up article would be appropriate:

Q: Why do you use a "boolean counting" variable?
A: I want to avoid counting objects that are being created as part of the counting process. My first approach was to store the objects in a fixed length array, but the problem is that before too long, that array would overflow. So, I want to tell the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

code: "Hold your horses, I'm going to count a bit". I hope that my code won't generate deadlocks, but I have a feeling that it might.

Q: How do you replace Object.class?
A: The way I used to do it in the past was to patch the rt.jar file. However, it is much easier to simply specify the path to the new Object.class file when you run your program. You can do that by calling java -Xbootclasspath/p:<path to where java/lang/Object.class is> Test You might find that this produces a security exception, someone has reported that from Weblogic. If it does, you might have to patch the jar file and resign it or something, I dunno.

Q: While the code you wrote helps find how many objects are created, your friend stated he wanted to know about memory leaks. The code does not indicate how many, of any object type, he has in memory at any one point in time. I was trying to work out how one would do that, thinking maybe you could override finalize to decrement the counter. But I suppose, unlike the constructor, you're not guaranteed of it being called. Do you think there might be a way to access some internal tables to find out what references exist, or something like that?
A: Something I tried, which kept me occupied for quite a while, was to implement finalize() in Object to decrease the counter. Sure, you still don't know what really needs to be on the heap, but if you call System.gc() a few times you should have a good chance. However, in JDK 1.3 and JDK 1.4 I got a core dump when I tried to put ANY code in the finalize() method of Object.

Q: I get error messages when I try to compile your Object file with Ant. What should I do?
A: Erik Hatcher: Ant definitely has its share of issues, and it might require you to do a fork="true" to spawn a new VM in order to compile something like Object as it has some convoluted and sometimes in-the-way classloader issues. Heinz Kabutz: Alternatively, compile by just calling javac directly.

In this country we used to celebrate the 1st *and* the 2nd of January as holidays. The 1st was a day of heavy drinking on the beach (with subsequent work for the lifesavers :-(, and the 2nd was the day to sleep off the babbelas (hangover) of the 1st. For some reason, our new government stopped having the 2nd as a holiday, but most pragmatic businesses still take the 2nd off. Most sensible folk (like me ;-) avoid the beach on the 1st, so today is a holiday for me without the unpleasantness of a hangover combined with a sunstroke.

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 039] - Why I don't read your code comments ...

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 39th edition of "The Java(tm) Specialists' Newsletter", sent to over 2400 Java experts in 72 countries, with the most recent addition being Costa Rica. In this newsletter I am talking rather philosophically about commenting your code, looking at some examples from the JDK. The newsletter is a bit different to my normal ones, but didn't require less work (don't all unsubscribe at once, ok ;-)

Mauritius - Paradise Island. Blue sands, crystal skies, white water. *yawn* - who wants beach anyway when you could be doing computer stuff? How about adding some *real* fun to the equation? How about having your cake *and* eating it too? How about a Design Patterns course at the colourful Coco Beach Hotel situated at Belle-Mare on the East coast of Mauritius? To find out more about this unique opportunity to spend your tax money on a holiday ;-), click on http://vik.cshons99.net/lifetimelearning.

Why I don't read your comments

Many years ago, my brothers and I clubbed together and bought ourselves a ZX Spectrum 48k. Nice computer, especially after I stuck it inside a DK Tronics keyboard, but the problem we had with our machine was that after a random delay, it would reset itself. My big brother subscribed to a computer magazine, which included listing for games, etc. for the ZX Spectrum. There was one particular game to do with some rocket, which wasn't too long, so I decided to type it in. Somewhere in the text it said that the game had been tested only on the 16KB Spectrum, but I thought: "What the hell, I'll try it out." After *many* resets, it was finally finished, and, guess what, it never worked! One day my brother was watching me type and noticed how I was typing in all the comments (prefixed by REM). I was of tender young age and couldn't understand how the computer could understand this English comment - maybe some internal fuzzy logic, or artificial intelligence? I was most disappointed when my brother told me that REM actually did exactly zilch. Nothing. Nada. Gar nichts. Tipota. At that moment, I developed a keen distrust of comments.

After spending some years programming in Basic (without comments) I enrolled in a university degree for Computer Science. One of the things the lecturers seemed to emphasize was that your code had to have comments. Infact, due to the meanness of their spirits, they used to give us bad marks if we didn't comment our code! So,

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

after one or two pracs with bad marks we all learned that it was a good thing to have comments. Good to have them so we wouldn't get bad marks. We then proceeded to work like this: Hack, hack, hack, hack, hack, hack. Oh, deadline time. Rename variable names from f, g, h to interest, bond, currency. Add comments before every line of code. For example (we had progressed to C at this point):

```
/* Conter variabble for "for" loop. */
int i;
/* Toatl of additions for calculaton */
int t;
/* Indicidual number for calclatuion */
int d;

/* "for" loop */
for (i=0; i<100; i++) { /* increment i by one until hunderd */
  d = f(); /* get ghe calue for d */
  t = t + d; /* ad it to t */
}
return t; /* return the variable t */
```

all the spelling mistakes in that code were intentional

Our lecturers, who only knew an old version of Algol, could now at least understand what we had coded, and guess what, we were rewarded for our diligence with fantastic marks. Our marks were allocated like this: Does it work correctly? 50% Was it commented? 100%

As you might imagine, through it all, I couldn't shake the feeling that I was wasting my time. Let's pan forward a few years to when I, rather naively still, started using Java. In those dark days, we didn't have fancy tools like intellisense, so I relied very much on JavaDocs. I thought JavaDocs were manna from heaven - at least we didn't have to waste our time converting our comments to documentation. Quite a few times though, I got into real trouble because I believed what was written in the JavaDocs of various libraries.

Some time after JBuilder 2, I started browsing to the source code of a class whenever I wanted to know something about it. I still use JBuilder 3, since it suits my needs very well, and whenever I need to know how to do something, I use either intellisense to remind me, or I look at the source code. It would never occur to me to read the comments that geeks like you and I had written "because it was the right thing to do". Why don't I like comments? Here are some of my reasons:

I know that in most cases, they were written by programmers who didn't believe in them.
There is a high probability that the comments do not adequately reflect what is really happening in the code.
Code that is written well should not need comments. There are exceptions - for example - interfaces need very detailed comments to define the various conditions. Sure, if you have method names f(), g(), h(), you will need to document what the methods do, but why not call them something sensible?
Instead of long pre- and post-conditions written in the comments, why not rather written extensive unit tests?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Some of these reasons are quite controversial, but I am in good company with my opinions. Let's qualify them a bit further by looking at some source snippets that made me snigger over the last few weeks.

Reminders - gotta fix this when I have time

The first one that made me laugh was found in JDK 1.3.1 in the java.awt.color.ColorSpace.getName(int) method. It had the classical *computer programmer under pressure* comment of "REMIND - ..."

From JDK 1.3.1: java.awt.color.ColorSpace:

```
/**
 * Returns the name of the component given the component index
 */
public String getName (int idx) {
    /* REMIND - handle common cases here */
    return new String("Unnamed color component("+idx+")");
}
```

I went back to JDK 1.2.2 and there was exactly the same comment (didn't have source to earlier versions on my disk). I immediately opened up the source for JDK 1.4 beta, and had to chuckle even more when I looked at this:

From JDK 1.4 beta: java.awt.color.ColorSpace:

```
/**
 * Returns the name of the component given the component index.
 * @param idx The component index.
 * @return The name of the component at the specified index.
 */
public String getName (int idx) {
    /* REMIND - handle common cases here */
    return new String("Unnamed color component("+idx+")");
}
```

Wow! So *that's* what idx meant! I thought it was some abbreviation of "idiots do xml". Oh, and getName() returns a name! Who would've thought that! Why didn't they rather spend more time fixing the code?

Another one that made me chuckle is java.util.Locale.toLowerCase(). Again we had a magical reminder - this time it said: "Look at optimizations later".

From JDK 1.3.1: java.util.Locale:

```
/*
 * Locale needs its own, locale insenitive version of toLowerCase to
 * avoid circularity problems between Locale and String.
 * The most straightforward algorithm is used. Look at optimizations later.
 */
private String toLowerCase(String str) {
    char[] buf = str.toCharArray();
    for (int i = 0; i < buf.length; i++) {
        buf[i] = Character.toLowerCase( buf[i] );
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    }
    return new String( buf );
  }
```

Sure this is inefficient, isn't the whole Locale class though? Even hashCode is synchronized - generally not a good idea! Why did the programmers not simply make it more optimal in the first place?

So, when do I like comments? I like comments when they provide information that I could not glean from the name of the method, such as pre/post conditions, etc. A place where comments are absolutely essential is with interfaces, since I cannot guess what the message would really do unless there was a good comment.

That's all for this week, I'm running out of time. Next week I will give you a more conventional newsletter again.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 040] - Visiting your Collection's Elements

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 40th edition of "The Java(tm) Specialists' Newsletter", sent to over 2550 Java experts in over 70 countries. Despite my plea that you don't unsubscribe, I had a rather surprising number of unsubscriptions, as programmers expressed their outrage at my audacity by voting with their feet. My views are my own *and* that of my employer - since I am my own employer ;-). I'm working on a program at the moment and I do make sure that our JavaDocs are up to date by running a Doclet

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

that tells me where I've forgotten a tag. Whenever I change a method, I nuke the comments, and then the Doclet tells me where I need to add a comment again.

The ideas in this newsletter were spawned by Inigo Surguy (inigosurguy@hotmail.com) who works in Lemington Spa in the United Kingdom. Inigo is the UK Head of Research and Development of Interactive AG. Inigo also pointed out BCEL to me, used to change byte code "on the fly". I will write about some application of that in future.

A few newsletters ago, I mentioned traffic fines, and that I had had only one speeding fine in all my life. Last Wednesday, I was on my way to a meeting with my auditor, I was late, and, hmmm, make that 2 traffic fines in all my life? The road between where I live and where my auditor works is notorious. The police tell you: "if you have a puncture on that road, please carry on driving slowly until you get to the next town. Don't worry about damaging your wheel - rather break your wheel than ..." Ok, I'm exaggerating a bit, but the point I'm making is that I had never seen a speed trap on that road, because the cops are too scared to hang around long enough to book you. Never, until last Wednesday. I was caught fair & square, doing 160km/h in a 120km/h zone. Fortunately, the cop was in a good mood, so we had a good laugh when he pulled over some cops who were speeding, and he kindly reduced my speed to 139km/h. The speeding fine ended up being ZAR 100, about US$ 8.50. I'd be quite interested to hear from you what type of punishment you would face in your country for getting caught doing 160km/h in a 120km/h zone ... [hk: in case there are any cops on this list, that story was purely ficticious :-]

Mauritius - Paradise Island. Blue sands, crystal skies, white water. *yawn* - who wants beach anyway when you could be doing computer stuff? How about adding some *real* fun to the equation? How about having your cake *and* eating it too? How about a Design Patterns course at the colourful Coco Beach Hotel situated at Belle-Mare on the East coast of Mauritius? To find out more about this unique opportunity to spend your tax money on a holiday ;-), click on http://vik.cshons99.net/lifetimelearning. There are only a few places left, so please don't miss this unique opportunity.

Visiting your Collection's Elements

The Problem

I'm getting tired. Not tired of writing newsletters, but tired of Java. Tired of writing the same code over and over again. For example:

```
// ...
Iterator it = ages.iterator();
while(it.hasNext()) {
  Integer age = (Integer)it.next();
  System.out.println("Now you're " + age +
    ", in 3 years time, you'll be " + (age.intValue() + 3));
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

I don't like that while loop with the iterator. Don't know *why* I don't like it, it just looks inelegant to me. I like the weird for loop for an iterator even less:

```
// ...
for(Iterator it = ages.iterator(); it.hasNext();) {
  Integer age = (Integer)it.next();
  System.out.println("Now you're " + age +
    ", in 3 years time, you'll be " + (age.intValue() + 3));
}
```

Lastly, I don't like downcasting and I don't like the problems that occur when you have different types in a collection.

A different way ...

Before looking at a solution, I would like to show how I would use iterators normally:

```
import java.util.*;

public class OldVisitingIteratorTest {
  public static void main(String[] args) {
    Collection c = new LinkedList();
    for (int i=0; i<3; i++) c.add(new Integer(i));

    Iterator it = c.iterator();
    while(it.hasNext()) {
      // lots of brackets - looks almost like Lisp - argh
      System.out.println(((Integer)it.next()).intValue() + 10);
    }

    c.add(new Float(2.1));
    c.add("Hello");

    it = c.iterator();
    while(it.hasNext()) {
      Object o = it.next();
      if (o instanceof Integer) {
        System.out.println(((Integer)o).intValue() + 10);
      } else if (o instanceof Number) {
        System.out.println(((Number)o).intValue() + 20);
      } else if (o instanceof String) {
        System.out.println(((String)o).toLowerCase());
      } else {
        System.out.println(o);
      }
    }

    it = c.iterator();
    while(it.hasNext()) {
      System.out.println(((Integer)it.next()).intValue() + 10);
    }
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

The output from that code is:

```
10
11
12
10
11
12
22
hello
10
11
12
Exception in thread "main" java.lang.ClassCastException: java.lang.Float
        at OldVisitingIteratorTest.main(OldVisitingIteratorTest.java:32)
```

Instead of constructing an Iterator and going through the Iterator and doing some operation on its contents, why not pass in an object with an execute() method that is called with each element? After some speed-typing yesterday, while waiting for my students at a Design Patterns course at the Strand Beach Hotel near Cape Town to finish an exercise, I came up with:

```
import java.util.*;
import java.lang.reflect.*;

public class VisitingIterator {
  /**
   * Ordering methods in "best-fit" order.
   */
  private static final Comparator METHOD_COMPARATOR =
    new Comparator() {
      public int compare(Object o1, Object o2) {
        Class paramType1 = ((Method)o1).getParameterTypes()[0];
        Class paramType2 = ((Method)o2).getParameterTypes()[0];
        return paramType1.isAssignableFrom(paramType2) ? 1 : -1;
      }
    };

  /**
   * Threadsafe version of visit.
   * @param lock the object on which to synchronize
   * @param task is an Object with an execute(...) : void method
   */
  public void visit(Collection c, Object task, Object lock) {
    synchronized(lock) {
      visit(c, task);
    }
  }

  /**
   * @param task is an Object with an execute(...) : void method
   */
  public void visit(Collection c, Object task) {
    TreeSet methods =  new TreeSet(METHOD_COMPARATOR);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    Method[] ms = task.getClass().getMethods();
    for (int i=0; i<ms.length; i++) {
      if (ms[i].getName().equals("execute")
          && ms[i].getParameterTypes().length == 1) {
        methods.add(ms[i]);
      }
    }
    Iterator it = c.iterator();
    while(it.hasNext()) {
      boolean found = false;
      Object o = it.next();
      Iterator mit = methods.iterator();
      while(!found && mit.hasNext()) {
        Method m = (Method)mit.next();
        if (m.getParameterTypes()[0].isInstance(o)) {
          try {
            m.invoke(task, new Object[] { o });
          } catch(IllegalAccessException ex) {
            // we were only looking for public methods anyway
            throw new IllegalStateException();
          } catch(InvocationTargetException ex) {
            // The only exceptions we allow to be thrown from
            // execute are RuntimeException subclases
            throw (RuntimeException)ex.getTargetException();
          }
          found = true;
        }
      }
      if (!found)
        throw new IllegalArgumentException(
          "No handler found for object type " +
          o.getClass().getName());
    }
  }
}
```

Instead of having that ugly while loop, we can now pass an object to the VisitingIterator and the correct execute(...) method is called for each element in the collection. The OldVisitingIterator now becomes:

```
import java.util.*;

public class VisitingIteratorTest {
  public static void main(String[] args) {
    Collection c = new LinkedList();
    for (int i=0; i<3; i++) c.add(new Integer(i));
    VisitingIterator vit = new VisitingIterator();

    vit.visit(c, new Object() {
      public void execute(Integer i) {
        System.out.println(i.intValue() + 10);
      }
    });
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    c.add(new Float(2.1));
    c.add("Hello");

    vit.visit(c, new Object() {
      public void execute(Object o) {
        System.out.println(o);
      }
      public void execute(Number n) {
        System.out.println(n.intValue() + 20);
      }
      public void execute(Integer i) {
        System.out.println(i.intValue() + 10);
      }
      public void execute(String s) {
        System.out.println(s.toLowerCase());
      }
    });

    vit.visit(c, new Object() {
      public void execute(Integer i) {
        System.out.println(i.intValue() + 10);
      }
    });
  }
}
```

The output from our new style is:

```
10
11
12
10
11
12
22
hello
10
11
12
Exception in thread "main" java.lang.IllegalArgumentException:
No handler found for object type java.lang.Float
        at VisitingIterator.visit(VisitingIterator.java:62)
        at VisitingIteratorTest.main(VisitingIteratorTest.java:33)
```

Perhaps I've been smoking Java for too long, but I much prefer that code to the while(it.hasNext()) ... but I have not had the chance to try this idea out "in the real world". I will start using it and let you know if it makes code neater (or not). I know that it will be less efficient, but then, Java is so slow anyway, I'd rather have cool style than super-optimal code.

Until next week ...

Heinz

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 041] - Placing components on each other

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 41th (or is it 41st?) edition of **The Java(tm) Specialists' Newsletter**, sent to 2690 Java experts in over 74 countries. This newsletter is for those that want to be able to stump their interviewer when they get tough Java interview questions. e.g. "Will this compile?"

```
public class FastString extends java.lang.String {
  private int hash = 0;
  public FastString(char[] value) {
    super(value);
  }
  public int hashCode() {
    if (hash == 0) {
      hash = super.hashCode();
    }
    return hash;
  }
}
```

Normal Java programmers would answer: "No.", but ardent followers of this newsletter will answer: "Have the JDK classes been modified in any way? With an unmodified JDK this would not compile, but if you made String non-final it would compile. You would also have to make it non-final in the runtime; otherwise you would get a runtime error. However, since JDK 1.3, it has not been necessary to cache the hash code in a subclass of String as that is done already inside String." This will so amaze the interviewer that you will have that hot job doing Java development in no time!

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Right now, I should've been in Mauritius presenting my Java course, and next week would've been my Design Patterns course. Of all the obstacles that courses face, this time we faced a French-style bureaucracy in Mauritius :-(Everything was organised: we had enough attendants, the hotel was booked, my flight was booked. However, we could not get the correct government approval in time, so I will have to wait a bit longer before I can go "teach in the sun" again. In the meantime I'm trying to learn that strange "sport" that all the sales people play - golf. (OK, I admit - I've been playing golf instead of writing newsletters *blush*)

Placing components on each other

Two weeks ago, I was presenting my course *Design Patterns - The Timeless Way of Coding* to some experienced Java developers, and we spent quite a bit of time arguing about the Composite pattern. To refresh your memory, **the** book on OO Design Patterns by Erich Gamma et al, contains the following classes in the structure section on Composite:

```
public abstract class Component {
  public void add(Component c) {}
  public void remove(Component c) {}
  public abstract void operation();
}
public class Leaf extends Component {
  public void operation() { /* do something */ }
}
public class Composite {
  private java.util.List children = new java.util.LinkedList();
  public void add(Component c) { children.add(c); }
  public void remove(Component c) { children.remove(c); }
  public void operation() {
    // or my cool VisitingIterator from last week ;-)
    java.util.Iterator it = children.iterator();
    while(it.hasNext()) {
      ((Component)it.next()).operation();
    }
  }
}
```

The question we were kicking around was: "Why are the *add()* and *remove()* methods in the top-level Component class?"

For an excellent discussion around this question, have a look at the Pattern Hatching column by John Vlissides in the September 2001 edition of Java Report.

Hoping to have whet your appetite, I will not pursue that discussion further here, but instead jump over to the JDK. A few years ago, I was looking at the Composite pattern and comparing it to the java.awt.Component. I found it interesting that java.awt.Component did not contain the add() and remove() methods, those were contained in the subclass java.awt.Container. However, when I looked at javax.swing.JComponent I noticed that it extended java.awt.Container and therefore contained the methods add() and remove().

The big question is: **Why?** Was it done like this because Java does not have multiple inheritance or was it done to more closely follow *the* Composite pattern? My bet is on

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

multiple inheritance being the reason, but if you have reliable information (not speculation) I'd love to hear from you.

So, how do you put this to use?

Multi-lined button

Say for example you want to have a JButton with multiple lines of labels on it. One way is to use HTML text, but then the default font is different to the normal JButton font. An answer is to stick a few JLabels on top of a JButton (remember that JButton extends AbstractButton extends JComponent extends Container). You can make such a button by doing the following:

```
import javax.swing.*;
import java.awt.*;

public class MultilineButton {
  public static void main(String[] args) {
    JButton button = new JButton();
    // It is important to set the layout manager of the button
    button.setLayout(new GridLayout(0, 1));
    // Then we simply add components to it!
    button.add(new JLabel("This is a", JLabel.CENTER));
    button.add(new JLabel("multiline", JLabel.CENTER));
    button.add(new JLabel("button.", JLabel.CENTER));
    JFrame f = new JFrame("Multi-line Button");
    f.getContentPane().setLayout(new FlowLayout());
    f.getContentPane().add(button);
    f.setSize(100,100);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.show();
  }
}
```

This will produce a button with several lines on it. The problem with that example is that the focus of the button is not shown.

Checkbox on a Button

What we can also do is put a JCheckBox (or any Component for that matter) on top of a JButton. This can be used to make *really* confusing user interfaces. Please don't actually do this, I'm just illustrating something here...

```
import javax.swing.*;
import java.awt.*;

public class CheckBoxOnButton {
  public static void main(String[] args) {
    JButton records = new JButton();
    records.setLayout(new BorderLayout());
    records.add(new JLabel("Show Records"), BorderLayout.NORTH);
    records.add(new JCheckBox("autoscroll"), BorderLayout.CENTER);
    JFrame f = new JFrame("CheckBox on Button");
    f.getContentPane().setLayout(new FlowLayout());
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    f.getContentPane().add(records);
    f.setSize(200,200);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.show();
  }
}
```

Not very useful? You've seen nothing yet!

Why not a Tree on a Button?

Of course, if we can add a JCheckBox to a JButton, why can we not add a JTree to a JButton? Here's an example of how you can do that:

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;

public class TreeOnButton {
  public static void main(String[] args) {
    JButton button = new JButton();
    button.setLayout(new BorderLayout());

    final JLabel buttonText = new JLabel("Press me",
      JLabel.CENTER);
    button.add(buttonText, BorderLayout.NORTH);

    JTree tree = new JTree();
    tree.addTreeSelectionListener(new TreeSelectionListener() {
      public void valueChanged(TreeSelectionEvent e) {
        buttonText.setText("Press for " +
          e.getPath().getLastPathComponent());
      }
    });
    button.add(tree, BorderLayout.CENTER);

    JFrame f = new JFrame("Tree on Button");
    f.getContentPane().setLayout(new FlowLayout());
    f.getContentPane().add(button);
    f.setSize(500,500);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.show();
  }
}
```

Well, there you go. I have not found any IDE's which support this functionality directly, and I would be surprised if there was such an IDE. You should not really add complex components to each other, as it will confuse your users. Imagine trying to write a user manual for buttons containing trees and check boxes!

This was again one of the funnest newsletters to write, I'm looking forward to your feedback :-)

Kind regards

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 042] - Speed-kings of inverting booleans

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 42nd edition of **The Java(tm) Specialists' Newsletter**, sent to 2779 Java experts in over 75 countries. I'm writing this newsletter about 30'000 feet above mother earth on my way to Mauritius. Yes, eventually we managed to sort out all the bureacratic problems, and had to just shift the Design Patterns course by one week :-)))

Speed-kings of inverting booleans

About 10 days ago, I was chatting on ICQ to Roman Porotnikov, the best Java programmer in the Ukraine according to Brainbench, when he posed an interesting question:

"What's more quick variant for flag = !flag;? :) (one guy said flg = !flg; is an answer ;))"

I didn't really know the answer, so I guessed: "Probably flag = flag ? false : true;"

Being the avid programmer that I am, I quickly wrote a test program:

```
public class NotTest1 {
  public static void main(String[] args) {
    boolean flag = true;
    long start;
    start = -System.currentTimeMillis();
    for (int i=0; i<100000000; i++) {
      flag = !flag;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
    start += System.currentTimeMillis();
    System.out.println("time for flag = !flag: " + start + "ms");

    start = -System.currentTimeMillis();
    for (int i=0; i<100000000; i++) {
      flag = flag?false:true;
    }
    start += System.currentTimeMillis();
    System.out.println("time for flag = flag?false:true: " + start + "ms");
  }
}
```

Imagine my glee when I saw the following performance results. Roman might be the best Java programmer in the Ukraine, but I am the best Java programmer on this airplane!

```
time for b = !b: 1712ms
time for b = b?false:true: 1132ms
```

I was still puzzling over this as I could not understand how that could possible be faster, when Roman piped up:

"The answer is actually flag ^= true;"

Hmmmm - XOR on a bitwise level - sneaky! I added his "way" to my test to see if it really was faster, although I did believe that bitwise manipulation should be faster, but you never know with Java ;-)

```
public class NotTest2 {
  public static void main(String[] args) {
    boolean flag = true;
    long start;
    start = -System.currentTimeMillis();
    for (int i=0; i<100000000; i++) {
      flag = !flag;
    }
    start += System.currentTimeMillis();
    System.out.println("time for flag = !flag: " + start + "ms");

    start = -System.currentTimeMillis();
    for (int i=0; i<100000000; i++) {
      flag = flag?false:true;
    }
    start += System.currentTimeMillis();
    System.out.println("time for flag = flag?false:true: " + start + "ms");

    for (int i=0; i<100000000; i++) {
      flag ^= true; // XOR
    }
    start += System.currentTimeMillis();
    System.out.println("time for flag ^= true: " + start + "ms");
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

And of course, Roman was right, as you can see from the figures below.

```
time for flag = !flag: 1722ms
time for flag = flag?false:true: 1162ms
time for flag ^= true: 781ms
```

Interesting figures. It *proves* that my version is 32% faster and that Roman's version is 55% faster.

I mentioned this strange idea to Paul van Spronsen and he suggested we look at the generated bytecode. You can disassemble Java bytecode with the javap tool that forms part of the JDK. [HK: at this point of writing, we hit some turbulence and our food was being served so I thought it best to wait until the hotel. I must just add that this is the best hotel I've stayed at in all my travels and we are planning another bunch of courses in May - will let you know next newsletter. Back to the newsletter ...] In order to be able to compare the bytecode easily, I've split the cases into Normal.java, Faster.java and Fastest.java.

```
public class Normal {
  public void test() {
    boolean flag = true;
    flag = !flag;
  }
}
```

Compiling this class and running the command javap -c Normal produced the following for method test (comments are mine):

```
Method void test()
   0 iconst_1   // push constant "true"
   1 istore_1   // store in location 1 (flag)
   2 iload_1    // load value in location 1
   3 ifne 10    // if value is false goto bytecode 10
   6 iconst_1   // push constant "true"
   7 goto 11    // goto location 11
  10 iconst_0   // push constant "false"
  11 istore_1   // store value on stack in location 1
  12 return     // duh - this is obvious
             // don't you just LOVE assembler comments?
```

Ok, that was fairly optimal... Let's look at the next case and see how it differs.

```
public class Faster {
  public void test() {
    boolean flag = true;
    flag = flag?false:true;
  }
}
```

The resultant bytecodes were:

```
Method void test()
   0 iconst_1   // push constant "true"
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
 1 istore_1   // store in location 1 (flag)
 2 iload_1    // load value in location 1
 3 ifne 10    // if value is true goto bytecode 10
 6 iconst_0   // push constant "false"
 7 goto 11    // goto location 11
10 iconst_1   // push constant "true"
11 istore_1   // store value on stack in location 1
12 return
```

Identical? Yep, pretty much identical. The only difference is in one case we are testing for "equal" and in the other we are testing for "not equal". Surely that could not make such a big difference? (I'll leave the decompiling and understanding of the XORoman way as an exercise to the reader ;-)

```
public class Fastest {
  public void test() {
    boolean flag = true;
    flag ^= true;
  }
}
```

What happened? I have to assume that some part of the hotspot kicked in after some iterations and that the second example was only faster because it was second, so I ran the examples longer:

```
public class Not {
  public static void test() {
    boolean flag = true;
    long start;

    start = -System.currentTimeMillis();
    for (int i=0; i<1000000000; i++) {
      flag ^= true;
    }
    start += System.currentTimeMillis();
    System.out.println("time for flag ^= true: " + start + "ms");

    start = -System.currentTimeMillis();
    for (int i=0; i<1000000000; i++) {
      flag = !flag;
    }
    start += System.currentTimeMillis();
    System.out.println("time for flag = !flag: " + start + "ms");

    start = -System.currentTimeMillis();
    for (int i=0; i<1000000000; i++) {
      flag = flag?false:true;
    }
    start += System.currentTimeMillis();
    System.out.println("time for flag = flag?false:true: " + start + "ms");
  }
  public static void main(String[] args) throws Exception {
    test();
    Thread.sleep(1);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    test();
  }
}
```

Letting it run longer certainly shows more truth:

```
time for flag ^= true: 12397ms
time for flag = !flag: 11356ms
time for flag = flag?false:true: 11326ms
time for flag ^= true: 5697ms
time for flag = !flag: 11326ms
time for flag = flag?false:true: 11326ms
```

We can learn two lessons from this:

`flag ^= true` is faster than `flag = !flag`
Never trust Java performance statistics.

Don't forget that an intelligent compiler could've recognised what you were doing and done it on a bit level. There are many factors that affect Java performance: architecture, compiler, hotspot compiler, hardware, etc. and these all play a role when it comes to determining performance.

That's all for tonight - even the mosquitos are asleep already so I better sign off.

Heinz

P.S. If you write to me, please feel free to address me as "Heinz" - we are casually formal in South Africa ;-)

P.P.S. Roman Porotnikov's ICQ number is 76669875 and he has an interesting webpage at http://ejb.how.to

---

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 043] - Arrgh, someone wants to kill me!

Author: Dr. Heinz M. Kabutz
Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 43rd edition of **The Java(tm) Specialists' Newsletter**, sent to 2885 Java experts in over 75 countries. This time, I am writing from a rather cold Deutschland (for my likings), though the inhabitants tell me it is actually quite warm again, and I get funny looks when I walk around wearing a thick jacket and a wooly hat. Whereas on South African roads, I am the speed king with my Alfa Romeo, here I drive in the slow lane (at 140km/h), so that the trucks can overtake me. Traffic is quite interesting here in the good ol' Germany. Did you know that since February, you are not allowed to use your indicators at a traffic circle in Germany?

Weshalb verbringe ich nun ein paar Wochen im kalten Deutschland, wenn ich doch den sch•nen Sommer in Kapstadt geniessen k•nnte? Gute Frage. In S˜dafrika gibt es viele hochleistende Entwickler, die erstens gute Qualit•t bringen und zweitens dazu auch noch preiswert sind. Mit der weltweiten Flaute ist es wichtig auf jeden Cent zu achten, ohne dabei Qualit•t zu verlieren. F˜r die n•chsten paar Wochen m•chte ich also versuchen Projekte innerhalb von Europa zu finden, die Java Experten ben•tigen und bei denen ich gute s˜dafrikanische Resourcen liefern kann. Ich bin bis Ende April in Deutschland, wenn Ihre Firma Projekte hat die Sie ausgeben m•chten, w˜rde ich Ihnen gerne einen Besuch abstatten, also schicken Sie mir doch bitte eine mail. Ich habe zus•tzlich noch ein paar offene Tage, an denen ich Java Beratung geben kann.

Whilst I am in Germany, I would like to organise a **Java_Specialist_Treff** get-together for my European readers so we can get the opportunity to meet each other. I will discuss some of my newsletters, we'll have some drinks together, basically have a nice, nerdy time ;-))) The meeting will be held in Karlsruhe, Baden-W˜rttemberg, close to the French border. I have not fixed a date and time yet, so if you would like to attend, please send me a quick email and tell me what dates until 28th of April would be most suitable for you.

Arrgh, someone wants to kill me!

Approximately a year ago, I wrote a newsletter about hooking into the shutdown call. In that newsletter, I rather snidely remarked that the shutdown process had some deficiencies, which rendered it useless for situations where you want to abort the shutdown process and go back to the program. Once you're in shutting down state, the program **will** die.

After I published that newsletter, a South African friend, James Pereira, sent me a code snippet that used OS signals directly for detecting when the program was being closed. You can use this mechanism to write your own shutdown hook mechanism (even for JDK 1.2), and put in whatever features you need.

The code that he sent me used some classes from the sun.* packages to achieve this. *Warning: The classes in the sun.* package hierarchy can be changed at the whim of Sun's fancy. In addition, you will probably not find a sun.* hierarchy in any VMs except Sun's, though I have not looked yet.* The package that James pointed me to was sun.misc, containing a flea-market (i.e. jumble-sale) of classes, some useful,

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

others badly designed. You've been warned - so please don't whinge if this does not work exactly like this on your IBM / MS / whatever VM.

When you press Ctrl+C in the command window, an OS signal is generated called SIGINT. This is the same on Windows and Unix. When you kill a program by pressing the little cross, or using your task manager, a SIGTERM signal is sent to your program, and it will not be back! What we can do is link into the signal handler and handle that signal any way we want to. For example:

```
import sun.misc.Signal;
import sun.misc.SignalHandler;

/**
 * The idea for this code came from James Pereira, this code
 * looks quite different to the well-structured, logically
 * named class he sent me.
 * The idea is that we register a handler for a Ctrl+C
 * signal and then handle it.
 */
public class Aaarggh {
  public static void main(String[] args) throws Exception {
    Signal.handle(new Signal("INT"), new SignalHandler () {
      public void handle(Signal sig) {
        System.out.println(
          "Aaarggh, a user is trying to interrupt me!!");
        System.out.println(
          "(throw garlic at user, say `shoo, go away')");
      }
    });
    for(int i=0; i<100; i++) {
      Thread.sleep(1000);
      System.out.print('.');
    }
  }
}
```

When you run this program, it will print dots onto the console and if you try to stop it with Ctrl+C, it will give you the following output and merrily carry on printing dots. You can either stop it by waiting for the 100 seconds or to kill it via the task manager. (Kill sends a SIGTERM signal, which is stronger than SIGINT)

```
.....Aaarggh, a user is trying to interrupt me!!
(throw garlic at user, say `shoo, go away')
..
```

Inside the signal handler you can do anything you want to, including calling System.exit(0) or Runtime.getRuntime().halt(0) if you don't want shutdown hooks to be called.

What signals can you catch? The answer to that question depends on the operating system you are running on. I managed to dig out a list for Windows and Solaris Sun JDK 1.2.2:

Windows: ABRT, FPE, ILL, INT, SEGV, TERM

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Solaris: ABRT, ALRM, BUS, CHLD, CONT, EMT, FPE, HUP, ILL, INT, IO, KILL, PIPE, POLL, PROF, PWR, QUIT, SEGV, STOP, SYS, TERM, TRAP, TSTP TTIN, TTOU, URG, USR1, USR2, VTALRM, WINCH, XCPU, XFSZ

Please don't ask me what events generate each of those signals. It's a miracle that I found that list of signals, sun.misc.Signal is written in such a way that if you pass in an incorrect String it throws an IllegalArgumentException at you. However, it does not tell you what the possible signals are. The most significant signals from within Java are SIG and TERM.

Here is some code that is called whenever our program is killed:

```
import sun.misc.Signal;
import sun.misc.SignalHandler;

public class SophisticatedShutdownSequence {
  private static boolean running = true;
  public static void init() {
    Runtime.getRuntime().addShutdownHook(new Thread() {
      public void run() {
        System.out.println("reached point of no return ...");
      }
    });

    SignalHandler handler = new SignalHandler () {
      public void handle(Signal sig) {
        if (running) {
          running = false;
          System.out.println("Signal " + sig);
          System.out.println("Shutting down database...");
        } else {
          // only on the second attempt do we exit
          System.out.println(" database shutdown interrupted!");
          System.exit(0);
        }
      }
    };
    Signal.handle(new Signal("INT"), handler);
    Signal.handle(new Signal("TERM"), handler);
  }

  public static void main(String args[]) throws Exception {
    init();
    Object o = new Object();
    synchronized (o) {
      o.wait(10000);
    }
    System.exit(0);
  }
}
```

If you run this and press Ctrl+C twice, or try kill the program once via task manager or by pressing the cross, you get the following output:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Signal SIGINT
Shutting down database...
 database shutdown interrupted!
We have now reached the point of no return ...

This shutting down sequence is a good opportunity for the State Pattern (see Gang of Four book). Due to popular request, I will try and show one Java Design pattern every 4 newsletters, and I might just start with the State Pattern. In addition, I will soon start doing book reviews on books that you should definitely have in your bookshelf.

Switching off OS signals at runtime

In JDK 1.3.1, Sun sneaked in a JVM runtime parameter to stop all this nonsense of hooking into signals.

```
public class NoShutdownHookAllowed {
  public static void main(String[] args) throws Exception {
    try {
      Runtime.getRuntime().addShutdownHook(new Thread() {
        public void run() {
          System.out.println("Shutdown hook going mad");
        }
      });
    } catch(IllegalArgumentException ex) {
      System.out.println("Caught " + ex);
    }
    Thread.sleep(10000);
  }
}
```

Try running this with JDK 1.3.1 and beyond, and use the following command: java -Xrs NoShutdownHookAllowed The -Xrs setting reduces use of OS signals by Java/VM.

What happens is that an IllegalArgumentException is generated after the thread has been added as a shutdown hook. This means that if you try to interrupt the code after it has been started, the shutdown hook will **not** be called. However, if you let the program finish naturally, it will run the shutdown hook. As far as I know, the -Xrs option is there to make it easier to write services in Java.

That's enough for now, please remember to send me an email if you would like to attend our **Java_Specialist_Treff** in Baden-W˜rttemberg during April, and tell me what dates would suit you best.

Also dann, bis n•chste Woche...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline
Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 044] - Review: Object-Oriented Implementation of Numerical Methods

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 44th edition of **The Java(tm) Specialists' Newsletter**, sent to 3101 (!) Java experts in over 77 countries. A special welcome to our first subscriber from Morocco!

I'm still in Germany, enjoying the technology (we won't mention the weather again, OK?) the people, the food, the fantastic beer. Not enjoying the shops that don't take credit cards. In South Africa I once bought a cabbage costing about US$ 0.13 on my credit card (long story, I thought I had some money in my wallet but it was completely empty).

Administrative Note: Before I get into this week's book review, there has been a slight change in the way this newsletter is going to be funded. I have purchased the rights to use an idea by Vince Sabio, now comfortably retired author of **HumourNet**, in order to enrich myself. He has patented a concept called an "Unsubscription Fee" for newsletters such as this. It's very simple really: Though subscriptions are free, unsubs cost US$5.00 to US$7.00, depending on your geographical location. The charge automatically appears on your credit card (when you unsubscribe) as "Maximum Solutions (South Africa)" (Please remember this when you get your bill.) This idea will also enable me to measure the percentage of intellectual proletariat (TM - Vince Sabio) on my distribution list.

Review: Object-Oriented Implementation of Numerical Methods by Dr. Didier Besset

As promised, this week I am going to look at an interesting book, that I think most of you will enjoy. The author is a subscriber to this newsletter, but that's no surprise, most of "who's who" of Java authors are on this list :-) My purpose in book reviews is to tell you of *interesting* and *different* books that I think you would appreciate.

In the days when I attended school we did not have computer studies, so our universities could not make that a prerequisite for Computer Science. The effect was that a wide variety of talent arrived at our hallowed halls of learning. This presented a problem for my old Computer Science Department: If they made CompSci too

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

difficult, then those who had never seen a computer would fail, and if they made it too easy, the hackers would get full marks, get bored, not go to class, etc. They therefore made the rule that you had to pass Mathematics II *before* taking CompSci III. The hardest three years for many a hacker was Mathematics II.

We learnt a whole lot of things at Mathematics, much of which I never fully understood or appreciated. I think that playing with computers was just so much more interesting than looking at a blank piece of paper. I find it much more interesting figuring out a computer program than reading mathematical proofs.

I wish I had had a copy of Dr. Didier Besset (PhD Physics from University of Geneva) book Object-Oriented Implementation of Numerical Methods in those days! It marries numerical methods and programming in a very interesting way. Just listen to these algorithms, implemented in Java *and* Smalltalk:

Interpolations: Lagrange, Newton, Neville, Bulirsch-Stoer and Cubic Spline
Zero of Function: Bisection Algorithm, Newton's Method, Roots of Polynomials
Integration of Functions: Trapeze, Simpson, Romberg
Series: Infinite Series, Continued Fractions, Incomplete Gamma Function, Incomplete Beta Function
Linear Algebra: Vectors, Matrices, and all that you might dream of for linear algebra
Elements of Statistics: Moments, Histograms, Random number generators, probability distributions
Statistical Analysis: Fisher-Snedecor, many others
Optimization: Hill-Climbing, Powell's Algorithm, Genetic Algorithm
Data Mining: Data Server, Covariance, Mahalanobis Distance, Cluster Analysis

It cuts, it slices, it dices! If you can't get excited by all those algorithms, then you're in the wrong profession ;-)

Dr. Besset sets the scene in the book by pulling out some performance statistics. We all know Java is slow ... right? Have a look at these stats from the book:

	Operation	Units	C	Smalltalk	
	Polynomial 10th degree	msec.	1.1	27.7	
	Neville Interpolation (20 points)	msec.	0.9	11.0	
	LUP matrix inversion (100 x 100)	sec.	3.9	22.9	

The C measures are done using published algorithms, so Dr. Besset didn't just add a whole lot of wait statements into the C code. Dr. Besset says: "I want to emphasize here that all the code in this book is real code that I have used personally in real applications." Wow, that's certainly better than most books nowadays :-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Comparing Doubles

Besides all the interesting algorithms, which are shown with mathematical explanations and well-written Java code, Dr. Besset also tackles issues such as the problems that happen when you compare floating point numbers. Here's an extract written by him for the Smalltalk Chronicles (edited by myself):

Dr. Didier Besset: "One classical caveat with floating-point numbers is checking the equality between two floating-point numbers. Now and then one bloke complains on some news groups that Smalltalk does not compute right with floating-point numbers. In the end it turns out that he was computing a result with method A, the same result with method B, and, to check the results, was evaluating the expression ' resultA == resultB. The fact that this expression evaluates to false has nothing to do with Smalltalk. It is a fundamental problem with floating-point numbers [HK: also in Java].

"A floating-point number is only an approximation of a mathematical real number. A small introductory article like this one is too short to explain things in depth, but I would like to quickly recall a few principles.

"Floating-point numbers are used to keep the relative error constant. This is valid of course for a given number. As soon as numbers are combined together one must follow the propagation of rounding errors. Because the relative error is kept constant, nothing serious happens with multiplications and divisions. The error on additions and subtractions, however, can become prohibitively large, up to the point of generating something utterly wrong. To illustrate this point, try running the following Java program:

```
public class DoubleTest {
  public static void main(String[] args) {
    System.out.println(2.71828182845905 - 2.71828182845904);
    System.out.println(2.71828182845905 - 2.71828182845904 +
      0.00000000000001 );
    System.out.println(2.71828182845905 == (2.71828182845904 +
      0.00000000000001));
  }
}
```

[HK: The answer is the following, a free unsubscription credit to anyone who guessed it... ;-]

```
9.769962616701378E-15
1.976996261670138E-14
false
```

"Mark the difference in the last digits! The result you will get is 100% wrong.

"Unless you are a very good and courageous mathematician, I would not recommend you to attempt to predict error propagation. The easiest and surest thing to do is to measure error propagation experimentally.

"After coding an algorithm, you can predict roughly where the infinities or the nearly zero cases are located. I am not speaking only about the result. All steps of the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

algorithm must be checked against the occurrences of infinities. In these areas, try to evaluate a few results by changing the values by a very small amount (10-12 or so for standard IEEE double format). In general the difference between the results will be one or two order of magnitudes larger than the original variation. If you observe something much larger, the algorithm used is not made for computers and must be adapted. I give several examples of such modifications in my book. Other examples can be found in Numerical Recipe for C ."

How should you compare Doubles?

Shortly after I started sending out my newsletter, a friend of mine mentioned to that he was surprised that Java programmers did not know how to compare doubles. If you just use "==" as in our example above, you will get incorrect results. Dr. Besset also has a section about that in his book. It is my understanding that in Java the precision of doubles and floats is defined by the IEEE 754 floating point format, so there should not be differences between physical architectures, since in Java we are running on a virtual machine. Please run these examples and tell me if your results vary. The results were identical on Wintel, AIX box running Java 1.3 on AIX version 4.3.3.0 (IBM 2 processor 4 cpu model 7044-270) and on a Solaris Box (SunOS Mars2 5.6 Generic_105181-29 sun4u sparc SUNW,Ultra-5_10).

Dr. Besset presents the following algorithm for comparing double precision numbers (reproduced with permission):

```
/**
 * This class determines the parameters of the floating point
 * representation
 * HK: I have refactored it somewhat to make it thread-safe and
 * to make it easier to understand and to fit into my newsletter.
 * The algorithms are the same as in the book.
 *
 * @author Didier H. Besset
 */
public final class DhbMath {
  /** Radix used by floating-point numbers. */
  private final static int radix = computeRadix();
  /** Largest positive value which, when added to 1.0, yields 0 */
  private final static double machinePrecision =
    computeMachinePrecision();
  /** Typical meaningful precision for numerical calculations. */
  private final static double defaultNumericalPrecision =
    Math.sqrt(machinePrecision);

  private static int computeRadix() {
    int radix = 0;
    double a = 1.0d;
    double tmp1, tmp2;
    do {
      a += a;
      tmp1 = a + 1.0d;
      tmp2 = tmp1 - a;
    } while (tmp2 - 1.0d != 0.0d);
    double b = 1.0d;
    while (radix == 0) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      b += b;
      tmp1 = a + b;
      radix = (int)(tmp1 - a);
    }
    return radix;
  }

  private static double computeMachinePrecision() {
    double floatingRadix = getRadix();
    double inverseRadix = 1.0d / floatingRadix;
    double machinePrecision = 1.0d;
    double tmp = 1.0d + machinePrecision;
    while (tmp - 1.0d != 0.0d) {
      machinePrecision *= inverseRadix;
      tmp = 1.0d + machinePrecision;
    }
    return machinePrecision;
  }

  public static int getRadix() {
    return radix;
  }

  public static double getMachinePrecision() {
    return machinePrecision;
  }

  public static double defaultNumericalPrecision() {
    return defaultNumericalPrecision;
  }

  /**
   * @return true if the difference between a and b is less than
   * the default numerical precision
   */
  public static boolean equals(double a, double b) {
    return equals(a, b, defaultNumericalPrecision());
  }

  /**
   * @return true if the relative difference between a and b is
   * less than precision
   */
  public static boolean equals(double a, double b, double precision) {
    double norm = Math.max(Math.abs(a), Math.abs(b));
    return norm < precision || Math.abs(a - b) < precision * norm;
  }
}
```

The book has details as to why the algorithms work the way they do. Here is how you would use the equals() method:

```
public class BetterDoubleTest {
  public static void main(String[] args) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    System.out.println("Floating-point machine parameters");
    System.out.println("--------------------------------");
    System.out.println("radix = " + DhbMath.getRadix());
    System.out.println("Machine precision = " +
      DhbMath.getMachinePrecision());
    System.out.println("Default numerical precision = " +
      DhbMath.defaultNumericalPrecision());
    System.out.println(DhbMath.equals(
      2.71828182845905,
      (2.71828182845904 + 0.00000000000001)));
    System.out.println(DhbMath.equals(
      2.71828182845905, 2.71828182845904));
    System.out.println(DhbMath.equals(
      2.718281828454, 2.718281828455));
    System.out.println(DhbMath.equals(
      2.7182814, 2.7182815));
  }
}
```

On the machines that I ran this test on, the output was:

```
Floating-point machine parameters
---------------------------------
radix = 2
Machine precision = 1.1102230246251565E-16
Default numerical precision = 1.0536712127723509E-8
true
true
true
false
```

That's it for this week, I hope you will consider these issues when next you want to compare doubles. And if you like interesting books, do yourself a favour and read Didier Besset's book. No, there's not really an unsubscription fee. Look at the date. Yes, I will get a referral fee if you purchase the book via that link.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 045] - Multi-line cells in the JTable

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 45th edition of **The Java(tm) Specialists' Newsletter**, read in over 78 countries, with newest additions Thailand and Iceland. Both end with "land" but they couldn't be two more opposite countries. Why don't those drivers who insist on crawling along the German Autobahn at 140km/h stick to the slow lane? I drove almost 500km on Friday, and had the opportunity to meet Carl Smotricz (my archive keeper) and some other subscribers in Frankfurt. We had some very inspiring discussions regarding Java performance, enjoyed some laughs at Java's expense and listened to my tales of life in South Africa.

Unsubscription Fees: Some of my readers wrote to tell me what a fantastic idea unsubscription fees were to make some money. Others wrote angry notes asking how I had obtained their credit card details. All of them were wrong! Note the date of our last newsletter - 1st April! Yes, it was all part of the **April Fool's** craze that hits the world once a year. Apologies to those of you who found that joke in poor taste (my wife said I shouldn't put it in, but I didn't listen to her). The rest of the newsletter was quite genuine. A friend, who was caught "hook, line & sinker", suggested that I should clear things up and tell you exactly what my purpose is in publishing "The Java(tm) Specialists' Newsletter":

#1. Publishing this newsletter is my hobby: No idealism here at all. A friend encouraged me a few years ago to write down all the things I had been telling him about Java, so one day I simply started, and I have carried on doing it. It's a great way to relax, put the feet up and think a while.

#2. There are no subscription / unsubscription fees: The day that I'm so broke that I need to charge you for reading the things I write, will be the day that I immediately start looking for work as a permanent employee again. There are neither subscription nor unsubscription fees, nor will there ever be.

#3. How *do* I earn my living? Certainly not by writing newsletters! I spend about 75% of my time writing Java code on contract for customers situated in various parts of the world. 20% of my time is spent presenting Java and Design Patterns courses in interesting places such as Mauritius and South Africa and the last 5% is spent advising companies about Java technology.

#4. Marketing for Maximum Solutions: Because people know my company and me through this newsletter, I have received many requests for courses, contract work and consulting, and this helps me to make a living. My hobby of writing the newsletter has turned out to have some nice side effects.

And now, without wasting any more time, let's look at a real-life Java problem...

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Multi-line cells in the JTable

The last slide of all my courses says that my students may send me questions *any* time they get stuck. A few weeks ago Robert Crida from Peralex in Bergvliet, South Africa, who came on my Java course last year, asked me how to display a JTextArea within a cell of a JTable. I sensed it would take more than 5 minutes to answer and being in a rush to finish some work inbetween Mauritius and Germany, I told him it would take me a few days to get back to him. When I got to Germany, I promptly forgot about his problem, until one of his colleagues reminded me last week.

Robert was trying to embed a JTextArea object within a JTable. The behaviour that he was getting was that when he resized the width of the table, he could see that the text in the text area was being wrapped onto multiple lines but the cells did not become higher to show those lines. He wanted the table row height to be increased automatically to make the complete text area visible.

He implemented a JTextArea cell renderer as below:

```
import java.awt.Component;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.table.TableCellRenderer;

public class TextAreaRenderer extends JTextArea
    implements TableCellRenderer {

  public TextAreaRenderer() {
   setLineWrap(true);
   setWrapStyleWord(true);
  }

  public Component getTableCellRendererComponent(JTable jTable,
      Object obj, boolean isSelected, boolean hasFocus, int row,
      int column) {
   setText((String)obj);
    return this;
  }
}
```

I wrote some test code to try this out. Before I continue, I need to point out that I use the **SUN JDK 1.3.1** whereas Robert uses the **SUN JDK 1.4.0**. The classic "write once, debug everywhere" is a topic for another newsletter ...

```
import javax.swing.*;
import java.awt.BorderLayout;

public class TextAreaRendererTest extends JFrame {
  // The table has 10 rows and 3 columns
  private final JTable table = new JTable(10, 3);

  public TextAreaRendererTest() {
   // We use our cell renderer for the third column
   table.getColumnModel().getColumn(2).setCellRenderer(
     new TextAreaRenderer());
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    // We hard-code the height of rows 0 and 5 to be 100
    table.setRowHeight(0, 100);
    table.setRowHeight(5, 100);
    // We put the table into a scrollpane and into a frame
    getContentPane().add(new JScrollPane(table));
    // We then set a few of the cells to our long example text
    String test = "The lazy dog jumped over the quick brown fox";
    table.getModel().setValueAt(test, 0, 0);
    table.getModel().setValueAt(test, 0, 1);
    table.getModel().setValueAt(test, 0, 2);
    table.getModel().setValueAt(test, 4, 0);
    table.getModel().setValueAt(test, 4, 1);
    table.getModel().setValueAt(test, 4, 2);
  }

  public static void main(String[] args) {
    TextAreaRendererTest test = new TextAreaRendererTest();
    test.setSize(600, 600);
    test.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    test.show();
  }
}
```

You'll notice when you run this, that when the row is high enough the text wraps very nicely inside the JTextArea, as in cell (0, 2). However, the JTable does not increase the row height in cell (4, 2) just because you decide to put a tall component into the cell. It requires a bit of prodding to do that.

My first approach was to override getPreferredSize() in the TextAreaRenderer class. However, that didn't work because JTable didn't take your preferred size into account in sizing the rows. I spent about an hour delving through the source code of JTable and JTextArea. After a lot of experimentation, I found out that JTextArea actually had the correct preferred size according to the width of the column in the JTable. I tried changing the getTableCellRendererComponent() method:

```
  public Component getTableCellRendererComponent(JTable jTable,
      Object obj, boolean isSelected, boolean hasFocus, int row,
      int column) {
    setText((String)obj);
    table.setRowHeight(row, (int)getPreferredSize().getHeight());
    return this;
  }
```

On first glimpse, the program seemed to work correctly now, except that my poor CPU was running at 100%. The problem was that when you set the row height, the table was invalidated and that caused getTableCellRendererComponent() to be called in order to render all the cells again. This in turn then set the row height, which invalidated the table again. In order to put a stop to this cycle of invalidation, I needed to check whether the row is already the correct height before setting it:

```
  public Component getTableCellRendererComponent(JTable jTable,
      Object obj, boolean isSelected, boolean hasFocus, int row,
      int column) {
    setText((String)obj);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    int height_wanted = (int)getPreferredSize().getHeight();
    if (height_wanted != table.getRowHeight(row))
      table.setRowHeight(row, height_wanted);
    return this;
  }
```

I tried it out (on **SUN JDK 1.3.1**) and it worked perfectly. There are some restrictions with my solution:

It will only work when only one column contains the TextAreaRenderer. You can easily write around this problem by having coordination between the various TextAreaRenderers.
You need to implement a TextAreaEditor, which I imagine will actually be a lot easier.

Satisfied, I sent off the answer to Robert, with the words: "After spending an hour tearing out my hair, I found a solution for you, it's so simple you'll kick yourself, like I did myself ;-)"

A few hours the answer came back: "Your solution does not solve my problem at all."

"What?" I thought. Upon questioning his configuration, we realised that I was using JDK 1.3.1 and Robert was using JDK 1.4.0. I tried it on JDK 1.4.0 on my machine, and truly, it did not render properly! What had they changed so that it didn't work anymore? After battling for another hour trying to figure out what the difference was and why it didn't work out, I gave up and carried on with my other work of tuning someone's application server. If you know how to do it in JDK 1.4.0, please tell me!

I have avoided JDK 1.4 for real-life projects, because I prefer others to find the bugs first. Most of my work is spent programming on real-life projects, so JDK 1.3.1 is the version I'm stuck with. My suspicion of new JDK versions goes back to when I started using JDK 1.0.x, JDK 1.1.x, JDK 1.2.x. I found that for every bug that was fixed in a new major version, 3 more appeared, and I grew tired of being a guinea pig. I must admit that I'm very happy with JDK 1.3.1, as I was with JDK 1.2.2 and JDK 1.1.8. I think that once JDK 1.4.1 is released I'll start using it and then you'll see more newsletters about that version of Java.

In a future newsletter I will demonstrate how you can implement "friends" at runtime in the JDK 1.4.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 046] - "The compiler team is writing useless code again ..."

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 46th edition of **The Java(tm) Specialists' Newsletter**, read in 82 countries, with newest additions of Guatemala and Bharain. After spending two months in Germany consulting on a project to get a 3-tier architectured program running faster (a lot faster), I am safely back in my own home in Somerset West, South Africa. Amazingly, I'm actually sticking to the speed limit, possibly because my Alfa makes as much noise at 100km/h as my uncle's Audi A6 made at 180km/h.

This newsletter has now been published in electronic book form (http://www.javafaq.nu/java/500/index.shtml) along with hundreds of tips on how to do complicated things in Java. The electronic book was put together by the author of JavaTips and is great for learning how to do clever things with Java. Please note that you do not need to get the electronic book to access my newsletters, they are still all available on my homepage.

During the two months in Germany, I learned a lot of new tricks for solving performance related problems, and they will be making great newsletter topics over the next few weeks, so stay tuned. This newsletter is actually about how much confidence Sun's JDK developers have in the work of their compiler developers' abilities.

Cloning vs. Serialization

One of the performance improvements that we did was write a cache for our business objects sitting in the middleware. It was meant to be a read-only cache, so we actually wanted to copy the business objects when we put them into the cache.

Since we wanted our framework to be easy for application developers to use, our first attempt was to avoid using the clone() method [clone does a shallow copy, and writing a correct deep clone() method is quite challenging]. Instead, we simply serialized the objects into the cache, and then deserialized them again on access. Our gut feeling was that this was a bad idea due to the performance problems with serialization, and after some tests we found out that cloning was about ten times faster then copying via the serialization mechanism.

A venture into unread(able) code

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

I was curious whether the serialization mechanism employed some clever tricks that I didn't know about, so I started digging a bit. I wonder how many have read the serialization code, specifically java.io.ObjectStreamClass? If you have, how far did you read? Right until the end?

Right at the bottom of the class java.io.ObjectStreamClass in the JDK 1.3.1 is the following code (this gem was removed in JDK 1.4):

```
  private final static Class[] NULL_ARGS = {};

  //WORKAROUND compiler bug with following code.
  //static final Class[] OIS_ARGS = {ObjectInpuStream.class};
  //static final Class[] OOS_ARGS = {ObjectOutpuStream.class};
  private static Class[] OIS_ARGS = null;
  private static Class[] OOS_ARGS = null;
  private static void initStaticMethodArgs() {
     OOS_ARGS = new Class[1];
     OOS_ARGS[0] = ObjectOutputStream.class;
     OIS_ARGS = new Class[1];
     OIS_ARGS[0] = ObjectInputStream.class;
  }
}
```

Higher up in the class we see where OIS_ARGS is used:

```
     /* Cache lookup of writeObject and readObject for
      * Serializable classes. (Do not lookup for
      * Externalizable)
      */
     if (serializable && !forProxyClass) {
        /* Work around compiler bug.  See declaration for
         * more detail.
         */
        if (OOS_ARGS == null || OIS_ARGS == null) {
           initStaticMethodArgs();
        }

        writeObjectMethod =
           getDeclaredMethod("writeObject", OOS_ARGS,
                 Modifier.PRIVATE, Modifier.STATIC);
        if (writeObjectMethod != null) {
           hasWriteObjectMethod = true;
        }
        readObjectMethod =
           getDeclaredMethod("readObject", OIS_ARGS,
                 Modifier.PRIVATE, Modifier.STATIC);
     }
```

What a strange piece of code? Why did the programmer write it like that? The reason for caching OIS_ARGS is obvious. It seems wasteful to construct a new class array each time you want to find the methods writeObject and readObject.

But what's this story about a compiler bug?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

We all make mistakes

Let's look more carefully at the declaration:

```
//WORKAROUND compiler bug with following code.
//static final Class[] OIS_ARGS = {ObjectInpuStream.class};
//static final Class[] OOS_ARGS = {ObjectOutpuStream.class};
```

I can picture the poor programmer tasked with producing the serialization mechanism in record time, typing ObjectInpuStream.class instead of ObjectInputStream.class then trying to compile his code, resulting in a message such as:

```
symbol  : class ObjectInpuStream
location: class ObjectStreamClass
static final Class[] OIS_ARGS = {ObjectInpuStream.class};
```

My first reaction would've been: "Oh no, those silly compiler guys cannot handle a circular dependency of classes within one package." I would then have had a coffee to destress and would've told all my colleagues that the compiler team were a bunch of boneheads who should've listened more carefully in their Compilers 101 class at university. I would also have realised that it was futile to try and convince "them" that they had made a mistake.

That was probably also the initial reaction of the Sun coder.

My second reaction would've been to initialize the static data members "lazily", that is, to write an init() method that is called at the first use. With that, I would've lost the "final" modifier on the data members.

That seemed to also have been the second reaction of the Sun coder.

My final reaction would've been to liberally sprinkle comments in my code to tell future readers that the compiler programmers are uncooperative boneheads who should rather be manually testing GUIs than writing compilers for a complicated language such as Java.

For some reason, I think that was also how our coder reacted.

I do not know how long it would have taken me to spot this simple spelling mistake. I know that if I counted all the hours that such small mistakes wasted, it would amount to quite a long time.

Lessons Learnt

Four eyes are better than two: Peer programming would have found this problem much quicker, and the programmer would not have been as easily tempted to churn out such code.

We all make mistakes: It is possible to write correct code that is poorly designed and inefficient. When something is slow or does not work, could it possibly be my fault?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Sun's JDK developers don't trust the ability of their compiler team: Should we?

Until the next issue...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 047] - Lack of Streaming leads to Screaming

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 47th edition of **The Java(tm) Specialists' Newsletter**, read by over 3600 Java programmers in 82 countries. I have put my Mauritius trip with photos on my website http://www.javaspecialists.co.za, please have a look under "Courses". **Warning:** prolonged looking at that webpage is known to cause envy - please enter at own risk!

I am planning Java and Design Patterns courses in South Africa for June. Please have a look at **our website** for more information.

Is your company thinking of venturing into new business but you do not have the necessary resources? Are you scared of subcontracting work to a non-English speaking country due to the communication problems involved? (no offense to my non-English readers - this is my advert, ok? ;-) Then South Africa is your dream come true. In South Africa we speak and write English fluently, so you will not have the typical communication problems that you would find in non-English speaking countries. Our software developers are *highly* skilled, very good at solving problems and able to pull up their sleeves and get to work. Internationally, South Africans are known for their hard work and dedication to the task at hand. If your company is

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

looking for such resources, please contact me by simply replying to this email, and I will personally see to it that you are contacted within 24 hours.

In the last two weeks I received two questions from readers who ran out of memory when trying to read a big object from the database. In this newsletter I want to explore how you can read a big object from a database without killing your poor JVM, and be scalable as well.

Lack of Streaming leads to Screaming

How do you retrieve big objects from the database in Java? Say you have a database containing previews of movies in DivX format, stored as IMAGE columns. How do you retrieve the 25 megabyte file from the database using JDBC?

Simple. We write a SELECT statement, execute it, and say result_set.getBytes(**1**). We run the code and it works well for small movie snippets, but as soon as we have a 25 megabyte file, our poor JVM throws an OutOfMemoryError. What's annoying about an OutOfMemoryError is that the stack trace is not filled in (because hey, you've run out of memory!), so you cannot exactly determine where the error occurred, unless you add trace logging. What makes it even more tricky is that some JDBC drivers try to be too clever, resulting in OutOfMemoryErrors.

Let's look at some test code. I have written two test classes, TestDatabaseBlobInsert and TestDatabaseBlobFetch. What surprised me was that of the several drivers that I tested (DataDirect, iNet SPRINTA, Avenir, MS SQL Server Type 4, JDBC/ODBC bridge), the JDBC/ODBC bridge was the fastest for inserting big objects into MS SQL Server. For fetching the data it was the slowest.

```
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.io.ByteArrayInputStream;

public class TestDatabaseBlobInsert {
  private static final String TABLE_DROP =
    "DROP TABLE MovieArchive";
  private static final String TABLE_CREATE =
    "Create Table MovieArchive (moviedata image, title varchar(255))";
  private static final String TABLE_INSERT =
    "INSERT INTO MovieArchive (title, moviedata) VALUES (?,?)";

  private static final int size = 25 * 1024 * 1024;
  private final byte[] data = new byte[size];

  private final Connection con;

  public TestDatabaseBlobInsert(String driver, String url,
      String user, String password)
      throws SQLException, ClassNotFoundException {
    Class.forName(driver);
    con = DriverManager.getConnection(url, user, password);
    System.out.println("Driver: " + driver);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    for (int i=0; i<data.length; i++)
      data[i] = (byte)(Math.random()*255);
  }

  public void setUp() throws SQLException {
    Statement st = con.createStatement();
    try {
      System.out.println("Dropping old table");
      st.executeUpdate(TABLE_DROP);
    } catch(SQLException ex) {} // table might not exist
    System.out.println("Creating new table");
    st.executeUpdate(TABLE_CREATE);
    st.close();
  }

  public void testInsertWithBinaryStream() throws SQLException {
    long start = -System.currentTimeMillis();
    System.out.println("Inserting via BinaryStream");
    PreparedStatement stmt = con.prepareStatement(TABLE_INSERT);
    ByteArrayInputStream bis = new ByteArrayInputStream(data);
    stmt.setString(1, "Babe");
    stmt.setBinaryStream(2, bis, data.length);
    stmt.executeUpdate();
    start += System.currentTimeMillis();
    System.out.println("That took " + start + "ms");
    stmt.close();
  }

  public void testInsertWithSetBytes() throws SQLException {
    long start = -System.currentTimeMillis();
    System.out.println("Inserting via setBytes()");
    PreparedStatement stmt = con.prepareStatement(TABLE_INSERT);
    stmt.setString(1, "On Her Majesty's Secret Service");
    stmt.setBytes(2, data);
    stmt.executeUpdate();
    start += System.currentTimeMillis();
    System.out.println("That took " + start + "ms");
    stmt.close();
  }

  public void testAll() throws SQLException {
    setUp();
    testInsertWithBinaryStream();
    testInsertWithSetBytes();
  }

  public static void main(String[] args) throws Exception {
    if (args.length != 4) usage();
    TestDatabaseBlobInsert test = new TestDatabaseBlobInsert(
      args[0], args[1], args[2], args[3]);
    test.testAll();
  }

  private static void usage() {
    System.out.println(
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      "Usage: TestDatabaseBlobInsert driver url username password");
    System.exit(1);
  }
}
```

I ran this code by setting up an ODBC source pointing to the MS SQL Server database Movies, and then running it with a maximum heap space of 256MB:

```
java -Xmx256m -classpath . TestDatabaseBlobInsert sun.jdbc.odbc.JdbcOdbcDriver
jdbc:odbc:Movies sa ""
```

The result on my little notebook was the following:

```
Driver: sun.jdbc.odbc.JdbcOdbcDriver
Dropping old table
Creating new table
Inserting via BinaryStream
That took 78975ms
Inserting via setBytes()
That took 73419ms
```

Back to the issue at hand - how do we get this data out of the database? The seemingly easiest way is to do the following:

```
1:  PreparedStatement st = con.prepareStatement(
  "SELECT moviedata FROM MovieArchive WHERE title = ?");
2:  st.setString(1, "Babe");
3:  ResultSet rs = st.executeQuery();
4:  if (rs.next()) {
5:    byte[] data = rs.getBytes(1);
}
```

This code can easily cause an OutOfMemoryError if the available heap memory is less than the size of the data that you are reading. Now for the 1'000'000 dollar question: Where does OutOfMemoryError occur? That depends on your driver. If you are using the iNet SPRINTA or the Avenir drivers, then you will run out of memory on line 4, i.e. when you call rs.next(). If you are using the DataDirect, Microsoft or ODBC bridge drivers, you will only get the out of memory error on line 5.

How can we write this so that we won't get an out of memory error? Here is some sample code. It is *very* important that you read the data blocks of bytes at a time, rather than in one big chunk, otherwise your system will definitely not scale to support many users.

```
import java.sql.*;
import java.io.*;

public class TestDatabaseBlobFetch {
  private static final String TABLE_SELECT =
    "SELECT moviedata FROM MovieArchive WHERE title = ?";

  private final Connection con;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public TestDatabaseBlobFetch(String driver, String url,
      String user, String password)
      throws SQLException, ClassNotFoundException {
    Class.forName(driver);
    con = DriverManager.getConnection(url, user, password);
    System.out.println("Driver: " + driver);
  }

  public void testSelectBlocksAtATime() throws SQLException {
    long start = -System.currentTimeMillis();
    System.out.println("SELECT: 64kb blocks at a time");
    PreparedStatement stmt = con.prepareStatement(TABLE_SELECT);
    stmt.setString(1, "Babe");
    ResultSet rs = stmt.executeQuery();
    int count=0;
    if (rs.next()) {
      try {
        System.out.println("Retrieving Data");
        OutputStream out = new BufferedOutputStream(
          new FileOutputStream("Data.1"));
        InputStream in = new BufferedInputStream(
          rs.getBinaryStream(1));
        byte[] buf = new byte[65536];
        int i;
        while((i = in.read(buf, 0, buf.length)) != -1) {
          out.write(buf, 0, i);
          count += i;
        }
        out.close();
      } catch(IOException ex) { ex.printStackTrace(); }
    }
    System.out.println("fetched " + count + " bytes");
    start += System.currentTimeMillis();
    System.out.println("That took " + start + "ms");
    stmt.close();
  }

  public void testSelectWithGetBytes() throws SQLException {
    long start = -System.currentTimeMillis();
    System.out.println("SELECT: all at once");
    PreparedStatement stmt = con.prepareStatement(TABLE_SELECT);
    stmt.setString(1, "Babe");
    ResultSet rs = stmt.executeQuery();
    byte[] data = null;
    if (rs.next()) {
      System.out.println("Retrieving Data");
      data = rs.getBytes(1);
      try {
        FileOutputStream out = new FileOutputStream("Data.2");
        out.write(data, 0, data.length);
        out.close();
      } catch(IOException ex) { ex.printStackTrace(); }
    }
    System.out.println("fetched " + data.length + " bytes");
    start += System.currentTimeMillis();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    System.out.println("That took " + start + "ms");
    stmt.close();
  }

  public void testAll() throws SQLException {
    testSelectBlocksAtATime();
    testSelectWithGetBytes();
  }

  public static void main(String[] args) throws Exception {
    if (args.length != 4) usage();
    TestDatabaseBlobFetch test = new TestDatabaseBlobFetch (
      args[0], args[1], args[2], args[3]);
    test.testAll();
  }

  private static void usage() {
    System.out.println(
      "usage: TestDatabaseBlobFetch driver url username password");
    System.exit(1);
  }
}
```

I tried this with several JDBC drivers, the only Type 4 driver that worked correclty was the DataDirect driver (now released under the Microsoft label). I will not go into the differences between the Microsoft driver and the others, that's for another article. If you want to try this out, you can run it like this:

```
java -Xmx2m -classpath .;msbase.jar;mssqlserver.jar;msutil.jar
  TestDatabaseBlobFetch com.microsoft.jdbc.sqlserver.SQLServerDriver
  jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=Movies sa ""
```

Naturally you have to download the Microsoft SQL Server Type 4 driver and put the jar files into the directory from which you are running this code. The output from using the DataDirect Microsoft driver is the following on my machine:

```
Driver: com.microsoft.jdbc.sqlserver.SQLServerDriver
SELECT: 64kb blocks at a time
Retrieving Data
fetched 26214400 bytes
That took 62746ms
SELECT: all at once
Retrieving Data
Exception in thread "main" java.lang.OutOfMemoryError
      <<no stack trace available>>
```

The iNet SPRINTA driver falls over much sooner - actually when you call rs.next():

```
Driver: com.inet.tds.TdsDriver
SELECT: 64kb blocks at a time
Exception in thread "main" java.lang.OutOfMemoryError
      <<no stack trace available>>
```

What is wrong here?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

I am finding it very hard to think of a reason to store 25mb files in a database. They are too big to stay in the database's cache for very long. I think that the design is flawed to start with. I would personally rather store the URL to the file in the database, instead of the actual data, and then retrieve the data directly from the file system.

I don't know all the conditions why someone would want to do that, but just remember, you have to stream such big data out of the database chunk by chunk, otherwise you have a serious problem.

Until the next issue ...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 048] - Review: The Secrets of Consulting

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 48th edition of **The Java(tm) Specialists' Newsletter**, sent to 3661 Java experts in over 82 countries. In this newsletter you are not a number, you are more than that. Yes, every one of my subscribers is an email address ;-) This week's newsletter is again rather philosophical - please excuse me, all my life I wanted to be a Philosopher :-)

After my traffic saga at the beginning of this year, I had the pleasure this morning of sorting out the shrapnel. I had sent a cheque to the provincial traffic department listed on the fine. One month later (I kid you not), the cheque was processed by the department and they discovered that the cheque was not bank guaranteed (as if I would try and rip off the police by sending a cheque that was not covered?!). Instead of writing a letter telling me about it, they sent me a registered letter two and a half months later, which I received this morning, telling me that I had missed my court appearance. The disturbing thing about it is that this letter was sent more than 100

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

days after I had sent the original cheque. So, my fine has now been tripled because I now also have a "contempt of court" fine. This morning I had to pick up the pieces, waiting for about 20 minutes to get through to the accounts department of the court. When I finally got through, I was told that they cannot give me my account status as they do not have a computer, but I could gladly make an appointment to come in and speak to the magistrate. This morning was spent writing a letter of protest and getting a bank guaranteed cheque from the bank, so I hope that I will not hear from them again.

Is your company thinking of venturing into new business but you do not have the necessary resources? Are you scared of subcontracting work to a non-English speaking country due to the communication problems involved? (no offense to my non-English readers - this is my advert, ok? ;-) Then South Africa is your dream come true. In South Africa we speak and write English fluently, so you will not have the typical communication problems that you would find in non-English speaking countries. Our software developers are *highly* skilled, very good at solving problems and able to pull up their sleeves and get to work. Internationally, South Africans are known for their hard work and dedication to the task at hand. If your company is **looking** for such resources, please contact me by simply replying to this email, and I will personally see to it that you are contacted within 24 hours.

Review: The Secrets of Consulting by Gerald M. Weinberg

A Guide to Giving & Getting Advice Successfully

The book that I want to write about this week is not really a Java book. Infact, it was written about ten years before Java was first released. However, this is a fantastic book on giving advice, receiving advice and getting your advice accepted. I have mentioned this book to many people, but no-one that I have spoken to had heard of it. I therefore have to conclude that the gems contained in this book truly are still "secrets". Please write to me if you have read the book.

Why would I even bother talking about books that have absolutely no relation to Java? The reason is simple: I have asked some of you what problems relating to Java occur in your companies. The majority of problems occur because of people, rather than technology. You might be highly enthusiastic about Kent Beck's ideas in eXtreme Programming Explained, but your colleagues and manager are more interested in "getting the job done" than in fancy new agile methodologies. (Not that the ideas are new, many of them were described in The Psychology of Computer Programming in 1971, the year in which I was born. Actually, the majority of our newsletter authors were born in 1971 - a good crop ;-) You might want to promote Java, but your colleagues are stuck with C++. Perhaps you think that Design Patterns are the way to go, but you are the only one in the company who thinks that.

This book is going to help you. It is going to help you to effectively give advice to your colleagues, managers and customers. I discovered this book by accident. About two years ago, Amazon recommended the book to me (the way they usually recommend irrelevant books). I would love to say that I immediately bought it, but I waited, and by the time I got to wanting to purchase the book, Amazon had no more stock. After much trouble, I eventually got my hands on a copy, and believe me, only my very best friends get to read it. I usually require their cars as deposit so that I will get the book back ;-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

What is really great about this book is that Weinberg presents his advice with stories, so each advice has a name that is very easy to remember. Here are some extracts from the book, shortened to save space. All quotes are in *italics*:

Prescott's Pickle Principle

"... My grandpappy used to tell a story about a stubborn cucumber. When he put it in the barrel, it looked around at the other cukes and was revolted by what had happened to them. 'Dadgum it,' he cursed. 'What's the matter with you guys? Have you no pride? No self-respecting cucumber would let himself get pickled without putting up a fight.'

"'But what can we do?' they'd ask. 'You can resist, that's what you can do. That's what I'm gonna do. No brine is going to get under my skin.'

Then grandpappy would stop, and I would always ask him, 'What happened to the stubborn cucumber?'"

"And what did he say?" I asked.

"He said, 'Don't be foolish, boy. If you stay in the brine long enough, you become a pickle.'"

i.e. ***Cucumbers get more pickled than brine gets cucumbered.***

i.e. ***A small system that tries to change a big system through long and continued contact is more likely to be changed itself.***

I like this advice. It tells me that if I hang around too much as a consultant within a company, I will not be effective anymore. It tells me that if I am in a situation which is much bigger than me, chances are that I will be changed myself. I felt that happen to me when I was a permanent employee. With my first paycheque, I was very enthusiastic. However, as I stayed longer, I felt that I was changing. I was starting to bicker and moan about company policies. I was no longer researching new technologies, because "the company" was supposed to send me on training. As with **all** companies, when it came to actually going on courses, there was either no time or no money in the training budget.

I am not suggesting that leaving your company is the right step for you. Being an independent contractor is only for people with a specific mindset. Only do it if you don't like sleeping, or if you cannot live unless you are completely stressed out. For some interesting insights about various roles that we fulfill in life, have a look at Robert Kiyosaki's Cashflow Quadrant.

How about another one:

The Law of the Hammer

"The Child who receives a hammer for Christmas will discover that everything needs pounding."

How often have you been in the situation where a company had decided to use Rational Unified Process / UML / XP / whatever, and this single fact would sort out all

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

their problems? No matter how much you tell your boss that Java is not a good idea for the next generation of Quake, he just will not listen. He is holding the Java hammer. I am not saying that any of the methodologies I mentioned are bad, I'm just saying that you have to apply them in the correct context.

I find the term "Software Factory" extremely amusing. My father runs a drinking straw factory (yes, someone has to make them). On the one side of the process you have an extruder which pushes out hot plastic and on the other side the straws come out, hygienically wrapped in plastic film. Having spent many hours working in factories, I am quite aware of how a factory works, and for the life of me cannot visualise software being produced the way that a factory produces straws. OK, besides the heap of scrap plastic that gets generated every month, that is possibly relevant.

Weinberg gives us some laws about consulting that make a lot of sense. They seem quite controversial, especially the third one, but if you think about it a while, you'll realise he's telling the truth. In his book he also goes into a lot more detail of why his rules work.

Laws of Consulting

The First Law of Consulting: *In spite of what your client may tell you, there's always a problem.*

The Second Law of Consulting: *No matter how it looks at first, it's always a people problem.*

The Third Law of Consulting: *Never forget they're paying you by the hour, not by the solution.*

The Fourth Law of Consulting: *If they didn't hire you, don't solve their problem.*

Conclusion

Don't bother asking me if you can borrow my copy of this book.

This is one fantastic book for anyone who uses consultants, or for someone who gives advice. Amazon's customer have rated this book with an average of more than 4.5 stars. As I said earlier, this book has nothing to do with Java, but it has everything to do with people. The problems you will find most of the time are directly related to people problems, rather than Java. Just remember this the next time you are told that your multi-billion dollar company cannot afford to pay for a license of a profiler to help solve your memory leak problems.

One last one, that I found very valuable: ***Clients always know how to solve their problems, and always tell the solution in the first five minutes***. Try it out next time.

For those of you who like to purchase books via Amazon, you can get there directly via this link.

Heinz

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 049] - Doclet for finding missing comments

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 49th edition of **The Java(tm) Specialists' Newsletter**, sent to over 3700 Java experts in over 82 countries. This week I am going to have a fun time running my Design Patterns Course at **"The Shuttleworth Foundation (TSF)"**. In the unlikely case that you are unfamiliar with the name "Shuttleworth", it belongs to the first African in space, the second space tourist, Mark Shuttleworth. His space trip has inspired many young people of South Africa to strive in Science and Mathematics. TSF is a non-profit organization aimed at providing help to schools, for example, a school administration tool.

Some of my readers mentioned after newsletter 047 that they could not find my travel report of Mauritius. Please have a look at **http://www.javaspecialists.co.za/designpatterns/mauritius.html** for a direct link.

Times are tough, I don't know if you're feeling it as well? When I started contracting, I went for over 3 years without a single day when I didn't have programming or training work. All of a sudden, I found myself pottering around for two weeks at a time, scratching my head wondering where the next project would come from. Fortunately, I have put my time to good use and developed some courses, which have done *extremely* well. However, I'm not programming that much anymore, and that upsets me greatly. I feel far happier when I'm working on a program than if I have to run around trying to sell my courseware. It is not even that my hourly rate for programming is excessively high. It's not. I would love to hear from you if you have a project that could do with some outside help from a Java specialist, *especially* if it can be done remotely.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Doclet for finding missing comments

A few newsletter ago, I made some comments about the fact that I rarely read comments. The response from people was overwhelming. There were very few neutral voices about what I had said: I was called "childish", "inexperienced" from the one camp, and "wise", "at long last someone has the guts to say it" from the other camp. A small detail that readers from both camps missed, was that I never said that I don't **write** comments, I merely said that I don't **read** them ;-)

Why do I **write** comments?

The person who has to maintain my code may not share my enthusiasm for reading code in order to understand what I was doing.
I quite like explaining in my comments the *why* of what I was doing.
The JavaDocs are a great tool for producing API documentation.

My biggest frustration with JavaDocs is that it is so difficult to remember keeping all the comments up to date *all the time*. One of my readers in India shared the same frustration, so she wrote a comment checker Doclet. I used her Doclet whenever I was programming, but it wasn't really very object-oriented.

I spent some time last weekend refactoring the program so that the code would be more understandable. This is quite a long newsletter, because of all the code. I have not added comments to the "CommentChecker", you'll have to figure out yourself how it works :-)

We start with the main class called CommentChecker, called by the javadoc system. In this class, I find all the classes from the RootDoc and I run a ClassChecker against them.

```
import com.sun.javadoc.ClassDoc;
import com.sun.javadoc.RootDoc;

public class CommentChecker {
  public static boolean start(RootDoc root) {
    ClassDoc[] classes = root.classes();
    for (int i = 0; i < classes.length; i++) {
      new ClassChecker(classes[i]).check();
    }
    return true;
  }
}
```

Let's also have two test cases, a class with comments called GoodTest ...

```
/** This is a test class */
public class GoodTest {
  /**
   * Constructor used for something
   * @param i used for something
   */
  public GoodTest(int i) {}
  /**
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
   * No-args constructor for GoodTest.
   */
  public GoodTest() {}
  /** This is a good comment */
  private boolean good;
}
```

... and a class with invalid or missing comments called BadTest.

```
public class BadTest {
  public BadTest(int i) {}
  /**
   * @param someone means nothing
   * @return always true
   * @throws bla if something bad happens
   */
  public BadTest() {}
  /**
   * @return nothing at all!
   * @return nothing at all!
   * @throws Exception if nothing happens
   * @throws Exception if something happens
   */
  public void method1() throws NullPointerException, Exception {}
  private boolean bad;
}
```

In order to call this, we execute the following command. To also check private data members / functions, we add the -private option.

```
javadoc -private -doclet CommentChecker *Test.java
```

For the GoodTest class, there is no output to System.err (because no comments are missing!). Depending on your company standards, you can change the Doclet to, for example, insist on an **@author** tag in the JavaDocs. My comments in the GoodTest class are nonsense of course - they have no meaning! "In the real world", I would have more sensibly named classes than GoodTest and the comments would also add value to the class. The output from running this doclet is:

```
BadTest misses comment
BadTest.BadTest(int) misses comment
BadTest.BadTest(int) misses comment for parameter "i"
BadTest.BadTest() misses comment
BadTest.BadTest() has unnecessary return comment
BadTest.BadTest() parameter "someone" does not exist
BadTest.BadTest() has unnecessary comment for exception "bla"
BadTest.method1() misses comment
BadTest.method1() has unnecessary return comment
BadTest.method1() has multiple comments for exception "Exception"
BadTest.method1() is missing comments for exception "NullPointerException"
BadTest.bad misses comment
```

Oh, I haven't shown you the rest of the classes, of course! I just wanted to whet your appetite so that you'll read the rest of this newsletter. As you can see, the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

output from the Doclet can be really useful if you want to make sure that you (or your client) have added all the necessary comments.

The hierarchy for my checking classes is as follows:

```
Checker
  |
  +-ClassChecker
  |
  +-ExecutableChecker
  |   |
  |   +-ConstructorChecker
  |   |
  |   +-MethodChecker
  |
  +-FieldChecker
```

Let's have a look at the Checker superclass:

```
import com.sun.javadoc.Doc;

/**
 * Abstract superclass for checking a code component.
 */
public abstract class Checker {
  private final Doc doc;
  public Checker(Doc doc) {
    this.doc = doc;
  }
  public abstract void check();
  protected abstract String getDescriptor();
  protected final boolean isEmpty(String s) {
    return s == null || s.trim().length() == 0;
  }
  public void checkComments() {
    if (isEmpty(doc.commentText()))
      error("misses comment");
  }
  protected void error(String msg) {
    System.err.println(getDescriptor() + " " + msg);
  }
}
```

We keep a handle to "Doc", which we use to test whether this code element has any comments. We also have an abstract check() method, which will be implemented differently for each code element. Each code element has a descriptor that we use to display which element an error belongs to.

Next we look at the class that checks whether a class has adequate comments:

```
import com.sun.javadoc.*;

/**
 * Check whether the class has comments
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
 */
public class ClassChecker extends Checker {
  private final ClassDoc doc;
  public ClassChecker(ClassDoc doc) {
    super(doc);
    this.doc = doc;
  }
  protected String getDescriptor() {
    return doc.qualifiedName();
  }
  public void check() {
    checkComments(); // calls superclass
    checkConstructors();
    checkMethods();
    checkFields();
  }
  private void checkConstructors() {
    ConstructorDoc[] constructors = doc.constructors();
    for (int i = 0; i < constructors.length; i++) {
      new ConstructorChecker(this, constructors[i]).check();
    }
  }
  private void checkMethods() {
    MethodDoc[] methods = doc.methods();
    for (int i = 0; i < methods.length; i++) {
      new MethodChecker(this, methods[i]).check();
    }
  }
  private void checkFields() {
    FieldDoc[] fields = doc.fields();
     for (int i = 0; i < fields.length; i++) {
      new FieldChecker(this, fields[i]).check();
    }
  }
}
```

This leads us to have a look at the ExecutableChecker class, a superclass of checking the comments of methods and constructors. The only difference between methods and constructors (as far as we are concerned) is that the constructor may not have a **@return** tag.

BTW, on a slightly different note, did you know that the following code compiles? i.e. you can have a method with the same name as the class. It can happen quite easily that you mean to write a constructor, but being a diligent C++ programmer you add the **void** keyword before the "constructor", thus actually writing a method. I discovered this a few years ago when one of my Bruce Eckel "Handson Java" students did this accidentally.

```
public class A {
  public void A() {}
}
```

Back to the problem on hand, a checker for methods and constructors. Since the only difference in our checking has to do with the return value, we make an abstract

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

method called checkReturnComments(). I'll let you figure out the checkParametersForComments() and checkExceptionComments() methods yourself.

```
import com.sun.javadoc.*;
import java.util.*;

public abstract class ExecutableChecker extends Checker {
  protected final String descriptor;
  private final ExecutableMemberDoc doc;
  public ExecutableChecker(ClassChecker parentChecker,
      ExecutableMemberDoc doc) {
    super(doc);
    descriptor = parentChecker.getDescriptor() + '.' +
      doc.name() + doc.flatSignature();
    this.doc = doc;
  }
  protected String getDescriptor() {
    return descriptor;
  }
  public void check() {
    checkComments(); // calls superclass
    checkReturnComments(); // calls subclass
    checkParametersForComments();
    checkExceptionComments();
  }
  public abstract void checkReturnComments();
  private void checkParametersForComments() {
    ParamTag[] tags = doc.paramTags();
    Map tagMap = new HashMap(tags.length);
    for (int i = 0; i < tags.length; i++) {
      if (tagMap.containsKey(tags[i].parameterName()))
        error("parameter \"" + tags[i].parameterName()
            + "\" has multiple comments");
      else if (!isEmpty(tags[i].parameterComment()))
        tagMap.put(tags[i].parameterName(), tags[i]);
    }
    Parameter[] params = doc.parameters();
    for (int i = 0; i < params.length; i++) {
      if (tagMap.remove(params[i].name()) == null
          && !params[i].name().equals("this$0")) {
        error("misses comment for parameter \"" +
          params[i].name() + "\"");
      }
    }
    Iterator it = tagMap.keySet().iterator();
    while (it.hasNext()) {
      error("parameter \"" + it.next() + "\" does not exist");
    }
  }
  private void checkExceptionComments() {
    ThrowsTag[] tags = doc.throwsTags();
    Map tagMap = new HashMap(tags.length);
    for (int i = 0; i < tags.length; i++) {
      if (tagMap.containsKey(tags[i].exceptionName()))
        error("has multiple comments for exception \"" +
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
        tags[i].exceptionName() + "\"");
    else if (!isEmpty(tags[i].exceptionComment()))
      tagMap.put(tags[i].exceptionName(), tags[i]);
  }
  ClassDoc[] exceptions = doc.thrownExceptions();
  for (int i = 0; i < exceptions.length; i++) {
    if (tagMap.remove(exceptions[i].name()) == null)
      error("is missing comments for exception \"" +
        exceptions[i].name() + "\"");
  }
  Iterator it = tagMap.keySet().iterator();
  while (it.hasNext()) {
    error("has unnecessary comment for exception \"" +
      it.next() + "\"");
  }
 }
 protected void foundCommentsForNonExistentReturnValue() {
  error("has unnecessary return comment");
 }
}
```

Jetzt haben wir das schlimmste hinter uns. Ooops - sorry - when I am tired I sometimes revert to my mother language ;-) Let's have a look at the checker for the constructors. All we do is check whether there is a tag for **@return** and if there is, the checker complains.

```
import com.sun.javadoc.ConstructorDoc;

public class ConstructorChecker extends ExecutableChecker {
 private final ConstructorDoc doc;
 public ConstructorChecker(ClassChecker parent,
    ConstructorDoc doc) {
  super(parent, doc);
  this.doc = doc;
 }
 public void checkReturnComments() {
  if (doc.tags("return").length > 0)
    foundCommentsForNonExistentReturnValue();
 }
}
```

The checker for methods is only marginally more complicated than that for constructors:

```
import com.sun.javadoc.*;

public class MethodChecker extends ExecutableChecker {
 private final MethodDoc doc;
 public MethodChecker(ClassChecker parent, MethodDoc doc) {
  super(parent, doc);
  this.doc = doc;
 }
 public void checkReturnComments() {
  Tag[] tags = doc.tags("return");
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    if ("void".equals(doc.returnType().qualifiedTypeName())) {
      if (tags.length != 0) {
        foundCommentsForNonExistentReturnValue();
      }
    } else if (tags.length == 0 || isEmpty(tags[0].text())) {
      error("missing return comment");
    } else if (tags.length > 1) {
      error("has multiple return comments");
    }
  }
}
```

Lastly, the checker for fields. We don't need to worry about return types, parameters and exceptions, so we simply check that it has a comment at all.

```
import com.sun.javadoc.FieldDoc;

public class FieldChecker extends Checker {
  private final String descriptor;
  public FieldChecker(ClassChecker parent, FieldDoc doc) {
    super(doc);
    descriptor = parent.getDescriptor() + '.' + doc.name();
  }
  public void check() {
    checkComments();
  }
  protected String getDescriptor() {
    return descriptor;
  }
}
```

If you stick all these classes in a directory and point JavaDoc onto them, you can use them to check that you have put comments with each important element. What's really nifty is that you can decide at runtime whether to show only public/protected elements or also package private or private.

The way that I use this Doclet is to only release classes once no messages are generated by this CommentChecker. When I change a method significantly, I will generally delete the comment, and then the comment checker will remind me at the next build that I need to add a comment. Because I get reminded to add the comments before I get to release the code, I avoid the pitfall of only adding the comments several months after I wrote the code.

This Doclet has been very helpful to me, in that it made my code look "very professional" (I can't believe I'm saying that ;-).

Attention: A lot of readers ask me whether they are allowed to use the code in my newsletters for their own projects (without paying me). Yes, you may freely use the code in my newsletters (at your sole risk), provided that you have a reference and acknowledgement in your code to my newsletter webpage http://www.javaspecialists.co.za. However, if my newsletters have *really* helped you or you are using code from my newsletters (especially this one), you can gladly make a donation to **Feeding the 5000** a non-profit organisation for providing food

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

and clothing to the really really poor in the Helderberg, the area of South Africa where I live.

In my next newsletter, I am going to make you scratch your head. I am going to demonstrate that it is possible to make your compiler fail because of what is contained inside a comment.

Until then ...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 050] - Commenting out your code?

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 50th edition of **The Java(tm) Specialists' Newsletter**. Last week I casually mentioned that I was presenting my Design Patterns Course at "The Shuttleworth Foundation". I really thought I was. However, when I arrived at TSF, I discovered that I had gone to the wrong place :-(My booking had been with Thawte Consulting, the company that Mark Shuttleworth sold to Verisign, and TSF has nothing to do with Thawte. I had simply assumed, and assumption is one of the biggest failings of a programmer.

When I finally got there, I was amazed that all my students were there already, waiting for me. (The course was supposed to start at 8:30 and I got there at 8:20). The next day, when I arrived at 8:20 again (after missing my turnoff due to daydreaming), the students were *again* all there, waiting for me. WOW, I thought, these guys are *really* keen! The bubble burst when they told me that somehow they got the message that the course started at 8:00 instead of 8:30!

The course went extremely well, partly due to very bright students and partly because I am getting more comfortable presenting this new course. Last Monday at

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

2:00am I at long last *clicked* with the GoF Factory Method. If you think you understand the Factory Method (like I did for a few years), my bet is that you actually don't ;-) You might understand a different type of pattern commonly *known* as "Factory Method", but there are very few people who understand *the* Factory Method according to GoF.

Please let me know if your company would like some Design Patterns training, by simply sending me an email. I will personally contact you within 24 hours and we can take it from there.

Commenting out your code?

Something I dislike even more than useless comments is code that is commented out. When I work with code, instead of commenting it out, I delete it. That way I don't have to remember why I commented out broken code, and it makes my code *much* easier to maintain. It's like telling the truth: you don't have to remember what you said. However, this newsletter is not a rant about commented out code, it is about how even commented out code can cause compiler failures, and how commented out code can end up as part of your class.

I want to thank Clark Updike from http://www.jhuapl.edu for the ideas that sparked this newsletter. He picked it up while studying for the Sun Certified Java Programmer (SCJP) Examination, which I did not author, by the way. I know the guys at Sun can be quite evil with the SCJP, but I don't think they would go this far. The SCJP seems to be getting more difficult to keep pace with the immense number of Java developers arriving on the scene. When I wrote it, you basically had to show up to get 91%.

I promised you last week that I would show you some code that could not compile, even though the problem code was commented out. Let's first look at the original program:

```
public class A1 {
  Character aChar = new Character('\u000d');
}
```

Try compile it, and you will get an error, such as:

```
A1.java:2: illegal line end in character literal
  Character aChar = new Character('\u000d');
                                  ^
1 error
```

Imagine sitting with this code, and not getting it to compile. What would you do? You would probably comment out the offending line to try get it to compile:

```
public class A2 {
  // Character aChar = new Character('\u000d');
}
```

You compile it, and what do you get: 2 errors instead of the 1 that you had without the comment!

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
A2.java:3: unclosed character literal
}
^
A2.java:3: <identifier> expected
}
  ^
2 errors
```

But we can get even more confusing, have a look at class A3:

```
public class A3 {
  // Character aChar = new Character('\u000d{System.out.println("Hello");}
}
```

Amazingly, it compiles, but does it do anything? Let's have a look at A3Test.java:

```
public class A3Test {
  public static void main(String[] args) {
    new A3();
  }
}
```

The test compiles, and when we run it, the output is:

```
Hello
```

Say what? I'm sure that you know what happens. The unicode character '\u000d' gets converted to a newline character as part of the "preparation for compile". At some point, before the class A2 actually gets compiled, it looks like this:

```
public class A2 {
  // Character aChar = new Character('
');
}
```

Obviously that does not compile! Similarly, at some point, A3 looks like this:

```
public class A3 {
  // Character aChar = new Character('
{System.out.println("Hello");}
}
```

I wish I could say: There, you should not comment out code! However, I can only say that you will get strange effects if you use '\u000a' or '\u000d' in your code.

Until the next newsletter ...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 051] - Java Import Statement Cleanup

Author: Dr. Cay S. Horstmann

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 51st edition of **The Java(tm) Specialists' Newsletter** sent to over 3800 readers in 84 countries, with latest additions of Chile and **Kenya**. I always get excited when I see an African name on my new subscriber list, since at the moment I only have subscribers in Africa from: Egypt, Ghana, Kenya, Mauritius, Morocco, Namibia, Nigeria, South Africa and Zimbabwe. I say "umkelekile" (welcome in Xhosa) to my new subscriber from Kenya.

This newsletter attracts the elite of Java programmers, since we cover things that are not usually mentioned on Java newsletters. If you are a subscriber to this newsletter, it makes you part of the "elite" :-) A relatively new subscriber to my newsletter is Dr. Cay S. Horstmann, famous Java author in the "Core Java" series. Cay very kindly pointed me to an article that he wrote about a tool he created for cleaning up **import** statements. Seeing that I am an amateur writer, I did not dare "edit" the article in case I completely messed it up, so I am sending it to you as is, with just the syntax highlighting added and font changed (almost like some students [and lecturers] do at universities). You can have a look at the original article on http://horstmann.com/articles/ImportCleaner.html.

Before we listen to what Dr. Horstmann has to say about the topic, I would like to make a few of my own comments about the subject. Having code with unnecessary **import** statements looks unprofessional, but how do you keep them up-to-date? An IDE which does this very nicely is Eclipse. You click on the class and say "Organize Imports", and **bang!** it beautifies your import statements. I am sure there are other IDEs out there that can do the same. However, if you don't have such an IDE, the technique described in this newsletter is a great way of solving this problem.

After my last Design Patterns Course, one of my students sent me the following note: *"I really enjoyed the course. I origionally thought it was going to be about learning to draw UML diagrams. I was pleasantly surprised to discover the course was actually about different programming strategies. I learnt some very cool tricks*

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

and new way of aproaching the problems of designing a system. A must for those who wish to design maintainable systems." Please send me an email if your company would like to receive training in Design Patterns.

Java Import Statement Cleanup

Background

In the Java programming language, the names of classes that are defined inside packages always start with the package name. For example, the class name

java.awt.Rectangle

starts with the name of the package java.awt. As a convenience to programmers, the **import** mechanism can be used to reference classes without the package name. For example, it is tedious to write

```
java.awt.Rectangle box = new java.awt.Rectangle(5, 10, 20, 30);
```

You can use the more convenient form

```
Rectangle box = new Rectangle(5, 10, 20, 30);
```

provided you *import* the class.

Classes in the same package are automatically imported, as are the classes in the java.lang package. For all other classes, you must supply an import statement to either import a specific class

```
import java.awt.Rectangle;
```

or to import all classes in a package, using the *wildcard* notation

```
import java.awt.*;
```

Importing classes can lead to *ambiguities*. A class name is ambiguous if it occurs in two packages that are imported by wildcards. For example, suppose a program contains the imports

```
import java.awt.*;
import java.util.*;
```

The class name List is now ambiguous because there are two classes java.awt.List and java.util.List. You can resolve the ambiguity by adding a specific import of the class name:

```
import java.awt.*;
import java.util.*;
import java.util.List;
```

However, if you need to refer to both java.awt.List and java.util.List in the same source file, then you have crossed the limits of the import mechanism. You can use

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

an **import** statement to shorten one of the names to List, but you need to reference the other by its full name whenever it occurs in the source text.

The Problem

Ambiguities are unpleasant because they can arise over time, as libraries expand. For example, in JDK 1.1, there was no java.util.List class. Consider a program that imports java.awt.* and java.util.* and uses the name List as a shortened form of java.awt.List. That program compiles without errors under JDK1.1 but fails to compile in Java 2.

Therefore, the use of wildcards for imports is somewhat dangerous. However, importing each class can lead to long import lists that are tedious to manage, especially as code is moved from one class to another during development.

To illustrate this, consider the import list in one of my recent files.

```
import java.awt.*;
import java.awt.geom.*;
import java.io.*;
import java.util.*;
```

It turned out that I really needed

```
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.geom.Point2D;
import java.awt.geom.Rectangle2D;
import java.util.ArrayList;
```

(Apparently, the need for importing java.io.* had gone away at some point during the program's evolution)

Thus, a problem that Java programmers face is how to keep import lists up-to-date when programs change.

Potential Solutions

One time-honored solution of checking import lists is to comment out one line at a time until compiler errors go away. Naturally, that is tedious.

Another solution is to stop using import lists altogether and referencing the full class names at all times. Naturally, that too is tedious.

Several compilers emit lists of classes that they load as they compile a program. For example, if you run the compiler in the Sun J2SE SDK 1.4 with the -verbose option, you get a list such as

```
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/Font.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/Graphics2D.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/Stroke.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/font/FontenderContext.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/geom/Line2D.class)]
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/geom/Point2D.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/geom/Rectangle2D.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/util/ArrayList.class)]
[loading ./AbstractEdge.class]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/lang/Object.class)]
[loading ./Edge.class]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/io/Serializable.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/lang/Cloneable.class)]
[loading ./LineStyle.class]
[loading ./ArrowHead.class]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/lang/String.class)]
[checking SegmentedLineEdge]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/Graphics.class)]
[loading ./SerializableEnumeration.class]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/util/AbstractList.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/util/AbstractCollection.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/geom/Line2D$Double.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/Shape.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/text/CharacterIterator.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/lang/Comparable.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/lang/CharSequence.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/geom/Point2D$Double.class)]
[loading /usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/awt/geom/RectangularShape.class)]
[loading
/usr/local/j2sdk1.4.0/jre/lib/rt.jar(java/text/AttributedCharacterIterator.class)]
```

It would be an easy matter to write a script that transforms this output into a set of import statements. However, the output contains classes that don't actually need to be imported (such as CharSequence and AttributedCharacterIterator). These classes are loaded because some of the loaded classes depend on them. It is not clear (at least to me) how one can weed out the unneeded classes.

I used a different approach. I wrote a utility that harvests the *class file*. Unlike source files, class files never contain shortened class files. Even standard classes such as java.lang.String are referenced by their full names.

Class files contain the names of classes as well as field and method descriptors that contain class names (in an odd format, such as Ljava/lang/String;). To harvest the class names, one must know the layout of the constant pool and the field and method descriptors. The class file format is well-documented--see the references at the end of this document--and only moderately complex.

The ImportCleaner Program

The ImportCleaner program parses one or more class files, harvests the class names, removes the unnecessary ones, sorts the remaining ones, and prints out a list of import statements to System.out.

Since ImportCleaner parses class files, your source file must first be compiled (presumably with a less-than-optimal import statement set). Then run ImportCleaner on the class file, capture the output, and paste the import lines into your source file.

For example, to run ImportCleaner on its own class files, you use

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

java -jar importcleaner.jar ImportCleaner

(You can find the ImportCleaner class files by unzipping importcleaner.jar).

The result is this list of imports, printed to System.out:

```
import java.io.DataInput;
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FilenameFilter;
import java.io.IOException;
import java.io.PrintStream;
import java.util.Iterator;
import java.util.Set;
import java.util.TreeSet;
```

Typically, your next step is to capture that list of imports and paste it into your source file.

If your source file contains multiple top-level classes, then you need to list all of them on the command line. For example,

java -jar importcleaner.jar MyClass MyHelperClassThatIsDefinedInTheSameFile

However, inner classes are located automatically.

You can supply the name of a class in any of the following forms:

without suffix: `MyClass`
with `.class` suffix: `MyClass.class`
with `.java` suffix: `MyClass.java`

The ImportCleaner program strips off the suffixes and then looks for the file MyClass.class and all files of the form MyClass$*.class (for inner classes).

If your class file is located in a package, you need to invoke ImportCleaner from the base directory. For example, if you use the package com.mycompany.myproject, invoke ImportCleaner from the directory that contains the com directory. You can supply the package name in either of the following forms:

using periods as separators: `com.mycompany.myproject.MyClass`
using the separator of your file system: `com/mycompany/myproject/MyClass` (or \ on Windows)

Capturing the output is very easy if you use the shell mode in Emacs. Other good programming editors have similar features. Alternatively, you can redirect the output to a file:

java -jar importcleaner.jar *class(es)* ***> outputfile***

The program takes the following options:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

`-wildcard`: This option produces an import list with wildcards instead of individual class names.
`-keepall`: This option keeps all imports, including the ones from `java.lang` and the current package. The default is to suppress the `java.lang` package and the current package
`-usecalls`: This option harvests method calls, which is sometimes beneficial to guess local variable types (see the Limitations section below). On the other hand, method calls can lead to spurious imports. This happens when you supply a parameter of a subclass type or catch a return value in a variable of a superclass type. For example, your code may call the constructor `java.io.DataInputStream(java.io.InputStream)` and pass a parameter of type `java.io.FileInputStream`, like this:

```
DataInputStream in = new DataInputStream(new FileInputStream(file));
```

Harvesting the method call yields the spurious **import** java.io.InputStream. Such spurious imports are generally harmless but unsightly.

Download and Installation

The usual.

Download the file `importcleaner.jar`. With some browsers, you may need to right-click on the link and select a menu option such as "Save Link As".
Save it in your favorite location
Open a command shell
Run `java -jar` your/favorite/location/`importcleaner.jar` options class(es)

License

The program is distributed under the GNU General Public License. Source code and a copy of the license are contained in the JAR file.

Limitations

Unfortunately, *constants* are not included in the class files, so this utility will miss them. Typical examples are:

BorderLayout.NORTH
Color.red

Also, the types of *local variables* are not included in the class file. This sounds like a big problem, but fortunately, the same class or interface name is often used in a method or field descriptor as well. To illustrate the issue, consider this code:

```
public void draw(Graphics2D g2) {
  Stroke oldStroke = g2.getStroke();
  g2.setStroke(new BasicStroke());
  // . . .
  g2.setStroke(oldStroke);
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

ImportCleaner includes java.awt.Graphics2D because it appears in the method signature of the processed class. It includes java.awt.BasicStroke because of the constructor call. But by default it won't include java.awt.Stroke since there is no guaranteed reference for it in the source file.

A remedy is to recompile after pasting in the ImportCleaner output. Then look at the error messages and manually insert the missing imports. It sounds bad, but in actual practice it doesn't seem to be all that bothersome.

Another remedy would be to use fully qualified class names in this situation.

You may also wish to use the -usecalls option. That option harvests method calls. In our example, that option finds the java.awt.Stroke class from the ()Ljava/awt/Stroke; method descriptor of the Graphics2D.getStroke method. Harvesting method descriptors is not the default because it can lead to spurious imports. (See the description of the -usecalls option for more information.)

References

The class file format is documented as part of the Java Virtual Machine specification (http://java.sun.com/docs/books/vmspec/2nd-edition/html/VMSpecTOC.doc.html). I found the source code at http://www.mcmanis.com/~cmcmanis/java/dump/ helpful to gain a clearer understanding of the class file format.

ImportScrubber (http://importscrubber.sourceforge.net/) seems to follow a similar approach as my `ImportCleaner`, and it has a GUI. BCEL (http://jakarta.apache.org/bcel/) is a library for reading and modifying class files used by ImportScrubber.

Importifier (http://javadude.com/tools/importifier/) is yet another tool for the same purpose, but it can't handle default packages.

A tool available at http://www.componative.com/downloads/impcleandoc.phtml harvests the output of the Jikes compiler.

TIM (http://www.chive.com/tim.htm) is a commercial program for import management.

That's it for this week.

Kind regards

Heinz

P.S. Even though my newsletter template says that Maximum Solutions, South Africa, has the copyright on this newsletter, the copyright of the content of the article belongs to Dr. Cay S. Horstmann. Please contact him if you would like to publish this article anywhere.

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists'*
Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 052] - J2EE Singleton

Author: David Jones (Virgin Mobile USA)

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 52nd edition of **The Java(tm) Specialists' Newsletter** sent to almost 4000 Java experts in 84 countries. Any day now we will count over to the 4000 mark! In countries besides my home country South Africa I would now get a bottle of champagne ready, but South Africa is one of only 12 countries where you can actually drink the water from the tap, and we are ranked #3 in terms of tapwater in the world - so I think I'll just drink a glass of fresh clear delicious South African water to celebrate the landmark 4000th subscriber :-)

This week I am very pleased to welcome a new contributor, David Jones, onto my newsletter. David is based in San Francisco, working for Virgin Mobile USA as a J2EE Architecture consultant. If you have an extremely lucrative job in that area, for someone with 4 years solid Java experience, you are welcome to contact David, although I don't know if he is actively looking for new challenges or not ;-) I might add that David is actually of British descent, so he knows how to do his math**s** and draw with colo**u**r.

I know that this is not the first newsletter in the world to speak about the Singleton pattern and its role in J2EE. I do hope, however, that it will show a different angle to what others have done.

Next week I am presenting a Java course in Cape Town, there is only one place left (someone has asked, but not confirmed, and as you know, first-come, first-served). If you live in the Cape Town area, and you would like to attend this course, please have a look at our website.

J2EE Singleton

What is a Singleton?

The Singleton pattern became widely popular when it was included in the now famous "Gang of Four" Patterns book [hk: see the brochure of my design patterns

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

course for information about several patterns]. Its general objective is to make sure that only one instance of a specific object type exists.

The classic code snippet used to create a Singleton is as follows

```
public class Singleton1 {
  private static Singleton1 instance = null;
  private Singleton1() {}
  public static synchronized Singleton1 getInstance() {
    if (instance == null)
      instance = new Singleton1();
    return instance;
  }
}
```

Some examples of a Singleton use the "Double Checked Locking method" to create the Singleton.

```
public class Singleton2 {
  private static Singleton2 instance = null;
  private Singleton2() {}
  public static Singleton2 getInstance() {
    if (instance == null) {
      synchronized(Singleton2.class) {
        if (instance == null) {
          instance = new Singleton2();
        }
      }
    }
    return instance;
  }
}
```

This method of "optimizing" the synchronization block has been shown not to work.

Singletons are commonly found in C++ and J2SE applications. However their use in distributed J2EE environment becomes a lot more complex and subject to debate.

Synchronization

Lots of common enterprise requirements are implemented to some degree by the J2EE Container. One of these classic requirements is the need for multi-threading in enterprise applications. This is generally implemented by the use of a thread pool. Each client request coming in is assigned a worker thread to service the request. Since therefore any J2EE application by its nature is multithreaded, limiting the synchronization of threads is an important requirement.

"Synchronized" as shown is used on a Singletons getInstance method. This is to prevent the "==null" test which is not thread safe causing the creation of two Singleton objects, one of which will then end up being thrown away.

There is a way around the "synchronize" issue by using a "startup class" or a "startup Servlet". Since both these types of objects are initialized at container start

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

up it is possible to use them to initialize the Singletons you need. It is then safe to drop the "synchronized" from the "getInstance" method.

The following StartUpServlet initializes several Singletons in its init method. This approach relies on the <load-on-startup> tag being placed in the Web Application's web.xml deployment descriptor.

```
import javax.servlet.ServletException;
import javax.servlet.http.*;
import java.io.IOException;

public class StartUpServlet extends HttpServlet {
  public void init() throws ServletException {
    try {
      // Initialize the Singletons.
      // these would be defined in separate files
      EJBController.initInstance();
      CacheManager.initInstance();
      RMISingletonWrapper.initInstance();
      DAOFactory.initInstance();
      ServiceLocator.initInstance();
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
  public void doGet(HttpServletRequest request,
      HttpServletResponse response)
      throws ServletException, IOException {}
}
```

Clustering and RMI Singletons

Clustering is when you have J2EE containers that are running on different VMs talk to each other. Clustering is used to provide load balancing and fail over for J2EE clients.

The simple/local Singleton as shown is a non-distributed object. Therefore in a clustered environment you will end up with at least one Singleton object on each server. This of course may be ok for the design requirements.

However if the design is to have one Singleton for the cluster then a common approach is to implement a "pinned service". This refers to an RMI object that is only located on one container in the cluster. Its stub is then registered on the clustered JNDI tree making the object available cluster wide. This raises of causes one issue, what happens when the server containing the RMI Singleton crashes?

A Container in the cluster could try to bind a new RMI Singleton if it notices it is missing out of the JNDI tree. However this could cause issues if all the containers try to bind new RMI Singletons at the same time in response to a failure.

Generally at the end of the day RMI Singletons do tend to have the drawback that they end up as single points of failure.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

In the following code example a local Singleton is used to act as a Wrapper around a RMI object that is bound into the clusters JNDI tree.

```
import javax.naming.*;
import java.rmi.*;

public class RMISingletonWrapper {
  private static RMISingletonWrapper instance = null;
  private static String SINGLETON_JNDI_NAME = "RMISingleton";

  public static RMISingletonWrapper getInstance() {
    return instance;
  }

  // All methods in delegate the method call to the actual
  // Singleton that lives on the clustered JNDI tree.
  public void delegate() {
    try {
      RMISingleton singleton = getRMISingleton();
      singleton.delegate();
    } catch (Exception e) {
      // Could try and recover
      e.printStackTrace();
    }
  }

  // Locate the true Singleton object in the cluster.
  private RMISingleton getRMISingleton() {
    RMISingleton rmiSingleton = null;
    try {
      Context jndiContext = new InitialContext();
      Object obj = jndiContext.lookup(SINGLETON_JNDI_NAME);
      rmiSingleton = (RMISingleton)PortableRemoteObject.narrow(
        obj,
        Class.forName("examples.singleton.rmi.RMISingleton"));
    } catch (Exception e) {
      // Could try and recover
      e.printStackTrace();
    }
    return rmiSingleton;
  }
}
```

Distributed Singleton Caches

One of the most common usages of Singletons is as caches of data. This use has issue for non RMI Singletons in a clustered environment. Problems happen when you attempt to do an update to the cache. Since a Singleton instance exists on each Container any update to the cached data by one Singleton will not be replicated to the other Singletons that exist on the other Containers.

This issue can be resolved by the use of the Java Messaging API to send update messages between Containers. In this approach if an update is made to the cache on one Container a message is published to a JMS Topic. Each Container has a listener

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

that subscribes to that topic and updates its Singleton cache based on the messages it receives. This approach is still difficult as you have to make sure that the updates received on each container are handled in a synchronous fashion. JMS messages also take time to process so the caches may spend some time out of sync.

In the following simplistic implementation of a distributed Cache a CacheManager Singleton holds a Map of cached items. Items to be cached are placed in a CachItem object which implements the ICacheItem interface.

The CacheManager does not make any attempt to remove old items from the Cache based on any criteria like "Last Accessed Time".

```
import javax.jms.*;

public class CacheManager implements MessageListener {
  public static CacheManager instance = null;
  public static Map cache = new HashMap();

  private TopicConnectionFactory topicConnectionFactory;
  private TopicConnection topicConnection;
  private TopicSession topicSession;
  private Topic topic;
  private TopicSubscriber topicSubscriber;
  private TopicPublisher topicPublisher;

  private final static String CONNECTION_FACTORY_JNDI_NAME =
    "ConnectionFactory";
  private final static String TOPIC_NAME = "TopicName";

  public static void initInstance() {
    instance = new CacheManager();
  }

  public static CacheManager getInstance() {
    return instance;
  }

  public synchronized void addCacheItem(ICacheItem cacheItem) {
    CacheMessage cacheMessage = new CacheMessage();
    cache.put(cacheItem.getId(), cacheItem.getData());
    cacheMessage.setMessageType(CacheMessage.ADD);
    cacheMessage.setCacheItem(cacheItem);
    sendMessage(cacheMessage);
  }

  public synchronized void modifyCacheItem(ICacheItem cacheItem) {
    CacheMessage cacheMessage = new CacheMessage();
    cache.put(cacheItem.getId(), cacheItem.getData());
    cacheMessage.setMessageType(CacheMessage.MODIFY);
    cacheMessage.setCacheItem(cacheItem);
    sendMessage(cacheMessage);
  }

  public ICacheItem getCacheItem(String key) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    return (ICacheItem)cache.get(key);
  }

  private CacheManager() {
    try {
      InitialContext context = new InitialContext();
      topicConnectionFactory = (TopicConnectionFactory)
        context.lookup(CONNECTION_FACTORY_JNDI_NAME);
      topicConnection = topicConnectionFactory.createTopicConnection();
      topicSession = topicConnection.createTopicSession(
        false, Session.AUTO_ACKNOWLEDGE);
      topic = (Topic) context.lookup(TOPIC_NAME);
      topicSubscriber = topicSession.createSubscriber(topic);
      topicSubscriber.setMessageListener(this);
      topicPublisher = topicSession.createPublisher(topic);
      topicConnection.start();
    } catch (Exception e) {
      e.printStackTrace();
    }
  }

  public void onMessage(Message message) {
    try {
      if (message instanceof ObjectMessage) {
        ObjectMessage om = (ObjectMessage)message;
        CacheMessage cacheMessage = (CacheMessage)om.getObject();
        ICacheItem item =  cacheMessage.getCacheItem();
        interpretCacheMessage(cacheMessage);
      }
    } catch (JMSException jmse) {
      jmse.printStackTrace();
    }
  }

  private void interpretCacheMessage(CacheMessage cacheMessage) {
    ICacheItem cacheItem = cacheMessage.getCacheItem();
    if (cacheMessage.getMessageType()==CacheMessage.ADD) {
      synchronized (this) {
        cache.put(cacheItem.getId(), cacheItem.getData());
      }
    } else if (cacheMessage.getMessageType()==CacheMessage.MODIFY) {
      synchronized (this) {
        cache.put(cacheItem.getId(), cacheItem.getData());
      }
    }
  }

  private void sendMessage(CacheMessage cacheMessage) {
    try {
      Message message = topicSession.createObjectMessage(cacheMessage);
      topicPublisher.publish(message);
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

Class Loading

Containers tend to implement their own class loading structures to support hot deployment for J2EE components and class isolation WAR files.

Class isolation in WAR files means that all classes found in a WAR file must be isolated from other deployed WAR files. Each WAR file therefore is loaded by a separate instance of the Class loader. The purpose is to allow each WAR file have its own version of commonly named JSPs like "index.jsp".

If a Singleton class is located in several WAR files it will mean that a separate Singleton instance will be created for each WAR file. This may of course be ok for the required design but it is worth being aware of.

Common Implementations of a Singleton in J2EE

In conclusion I would like to talk about the three most common implementations of the Singleton pattern in J2EE I have come across. These are as Service Locators, Object Factories and Controllers.

Service Locators

A Service Locator allows you to cache objects (like EJB homes) that are located in the JNDI tree. JNDI look ups are very expensive so using a Singleton to cache these objects is a good idea. Since you never update these objects it does not matter that the cache is not distributed across a cluster.

In the following implementation of a ServiceLocator all lookups into the JNDI tree are cached into a Map. The ServiceLocator's getService method has a threading issue where it could place the JNDI Object into the map multiple times. This is due to the fact that the check to see if the object is already in the cache is not synchronized. The reason for dropping the synchronization is that it is more costly to do a "synchronization" than just ignoring an issue that if it happened would cause no real harm.

```
public class ServiceLocator {
  private static ServiceLocator instance = null;
  private Map cache = null;

  private ServiceLocator() throws NamingException {
    cache = Collections.synchronizedMap(new HashMap());
  }

  public static ServiceLocator getInstance()
      throws NamingException {
    return instance;
  }

  public static void initInstance() throws NamingException {
    instance = new ServiceLocator();
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public Object getService(String entityName)
      throws NamingException {
    Object home = this.getFromCache(entityName);
    if (home == null) {
      try {
        Context jndiContext = new InitialContext();
        home = jndiContext.lookup(entityName);
      } catch (NamingException e) {
        e.printStackTrace();
      }
      this.putInCache(entityName, home);
    }
    return home;
  }

  private void putInCache(String key, Object value) {
    cache.put(key, value);
  }

  private Object getFromCache(String key) {
    return cache.get(key);
  }
}
```

Object Factory

An Object Factory class allows you to create an object of a certain type based on some form of an identifier. A common use of a factory in J2EE is in the creation of Data Access Objects. DAO's are commonly used to abstract SQL usage into specific classes. Clients obtain a factory Singleton and pass in the specific identifier for the specific DAO they need. The Object Factory instantiates the object and then returns a common interface to the client.

To allow possible support of multiple databases, a simple trick that can be used is to append a specific identifier to the DAO's class name. The implementer of the DAO class can then provide a specific implementation of the DAO for each database type if required. [HK: This is covered in our Design Patterns Course :-)]

It is possible that the factory could be implemented as a bunch of static methods. A reason not to take the static approach is the fact that maybe at some point the caching of DAO's will be required. If caching is required than the factories implementation can be changed to support caching without changing the factories' interface.

```
public class DAOFactory {
  private static DAOFactory instance = null;
  private String nameAppender = null;

  private final String ORACLE_DAO = "OracleDAO";
  private final String SQLServer_DAO = "SQLServerDAO";

  private DAOFactory() {
    // Maybe get this from a properties file
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    nameAppender = ORACLE_DAO;
  }

  public static DAOFactory getInstance() {
    return instance;
  }

  public static void initInstance() {
    instance = new DAOFactory();
  }

  public IDAO getDAO(String daoName) throws DAOException {
    IDAO aDAO = null;
    try {
      aDAO = (IDAO)
        Class.forName(daoName + nameAppender).newInstance();
    } catch (ClassNotFoundException cnfex) {
      throw new DAOException(cnfex);
    } catch (IllegalAccessException ilaex) {
      throw new DAOException(ilaex);
    } catch (InstantiationException instex) {
      throw new DAOException(instex);
    } catch (Exception e) {
      throw new DAOException(e);
    }
    return aDAO;
  }
}
```

Controller

A Controller takes a request and based on some criteria directs that request to an object to be serviced. Since the container is running multiple threads and garbage collection is expensive it would be nice to have just one instance of the controller. A Singleton therefore is a good protocol independent Controller.

In the following implementation of an EJB Controller, the processRequest method takes a Request object that specifies the following; the Stateless EJBs Home interface class name, the Home interfaces JNDI name, the method name within the Stateless EJB to call and finally the list of parameters to pass in. The Response object contains the return object from the method call.

```
public class EJBController {
  private static EJBController instance = null;

  public static void initInstance() {
    instance = new EJBController();
  }

  public static EJBController getInstance() {
    return instance;
  }

  public Response processRequest(Request request) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    Parameter ret = null;
    Object[] args = null;
    Class[] argTypes = null;
    Response response = new Response();

    List methodParameters = request.getMethodParameters();
    String methodName = request.getMethodName();

    if (methodParameters != null) {
      int parametersCount = methodParameters.size ();
      args = new Object[parametersCount];
      argTypes = new Class[parametersCount];
      for (int i = 0; i < parametersCount; i++) {
        Parameter param = (Parameter) methodParameters.get(i);
        args[i] = param.getValue ();
        argTypes[i] = param.getType ();
      }
    }
    try {
      EJBObject remoteObjRef = locateRemote(request);
      Class targetClass = remoteObjRef.getClass();
      Method m= targetClass.getMethod (methodName, argTypes);
      Object returnObject = m.invoke(remoteObjRef, args);
      response.setReturnObject(returnObject);
    } catch (InvocationTargetException e) {
      Throwable e1 = e.getTargetException();
      e1.printStackTrace();
    } catch (Throwable t) {
      t.printStackTrace();
    }
    return response;
  }

  // Locate the home interface of the JNDI tree and call its
  // create method
  private EJBObject locateRemote(Request request) {
    String homeInterfaceName = request.getHomeInterfaceName();
    String jndiName = request.getJNDIName();
    EJBObject remoteObjRef = null;
    try {
      Object obj = ServiceLocator.getInstance().getService(
        jndiName);
      EJBHome home = (EJBHome) PortableRemoteObject.narrow(
        obj, Class.forName(homeInterfaceName));
      Method createMethod = home.getClass().getMethod(
        "create", new Class[0]);
      remoteObjRef = (EJBObject) createMethod.invoke(
        (Object)home, new Object[0]);
    } catch (Exception e) {
      e.printStackTrace();
    }
    return remoteObjRef;
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The Singletons presented here as common implementations of the pattern in J2EE can be thought of as either stateless or ones that at most control resources/objects that do not need to be cluster aware.

That was an excellent piece of writing, very clear, concise, to the point - Thank you very much, David.

Until the next newsletter...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 053] - Charting unknown waters in JDK 1.4 Part I

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 53rd edition of **The Java(tm) Specialists' Newsletter** sent to over 4100 Java Specialists in 85 countries. Our friendly archive host Carl Smotricz has been experiencing some problems with his service provider, so our archive was offline for a bit. In the meantime, I have added my own archive to my website, so in future please look at http://www.javaspecialists.co.za for old issues.

Ahh, the pressures of life. This month has been hectically busy. The first week I gave some Java training at the University of Stellenbosch IT department to teach their Natural programmers Java. Very enjoyable week. The second week was spent presenting a Java 2 Standard Edition course and last week we moved in to our new house. I'm now sitting in my new office at home overlooking False Bay, and wondering what gems I can dig up in the source code of the JDK 1.4 ;-) Please

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

arrange to visit me if you ever come to the Helderberg, a village about 50km outside of Cape Town.

Would you like to find out how the new features of JDK 1.4 Standard Edition work? (logging, new IO, etc.) I am holding a JDK 1.4 crash course in Cape Town on the **29th of August 2002**. There are only **14** places available, which will be allocated on a strict first-come-first-served basis. The cost is only R1995 incl VAT for this course. Prerequisites for this course are a fairly good understanding of the JDK 1.2 or 1.3, although the Sun Certified Java Programmer Examination is not a prerequisite ;-). As with all my courses, if you are not completely satisfied with the course I will refund you 100% of the fees. You can register by simply replying to this newsletter and telling me your name, email and mobile number.

Charting unknown waters in JDK 1.4 Part I

As I said in my opener to this newsletter, I started by just looking at the java.util.* package, without even the subpackages, and that kept me busy for most of the afternoon. I have not finished yet, but I already have enough material for several newsletters.

toString() was broken - did you know that?

In the collection classes all hell breaks loose when a collection contains itself and you try to call toString() on it. For example, consider the following code:

```
import java.util.Hashtable;

public class HashtableTest {
  public static void main(String[] args) {
    Hashtable ht = new Hashtable();
    ht.put("Heinz", "Kabutz");
    ht.put(ht, "all");
    ht.put("all", ht);
    ht.put(ht, ht);
    try {
      System.out.println(ht);
    } catch (StackOverflowError e) {
      System.out.println("Caused Stack Overflow Error!");
    }
  }
}
```

Under the JDK 1.3 and before, when you ran this program, a Stack Overflow was generated, however, in JDK 1.4, the output is:

```
{Heinz=Kabutz, all=(this Map), (this Map)=all, (this Map)=(this Map)}
```

This same principle applies to the other collections. Is it a good idea to use a Hashtable as a key? Remember what happens when the hash code of a key changes during its life? I discussed that at length in Issue 031 of **The Java(tm) Specialists' Newsletter**. Each time a Hashtable changes, its hash code changes as well, so you should not use it as a key. However, there is no limit to the amount of stupid code

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

that gets written every day, and the chaps at Sun didn't expect people to add a collection to itself.

RandomAccess

A highly paid software developer once wrote something along these lines:

```
/** @author Mr M.O.Nument */
import java.util.*;
public class ListSearching {
  private List names = new LinkedList();
  public void f() {
    for (int i=0; i<size(); i++) {
      if (names.get(i) == "Heinz")
        System.out.println("Found it");
    }
  }
  private int size() {
    int result = 0;
    Iterator it = names.iterator();
    while(it.hasNext()) {
      result++;
      it.next();
    }
    return result;
  }
}
```

Take a moment to read through the code and try to understand it. In defense of the programmer, firstly he had not been on my Java course ;-), secondly there was a lot of code inbetween f() and size() so the problem was not as obvious as we can now see and thirdly the list was always very short so performance was not an issue either.

In JDK 1.4, a new tag interface (like java.io.Serializable) was introduced into the java.util package, called RandomAccess. This allows classes such as Collections to optimize their algorithms in the case where:

```
for (int i=0, n=list.size(); i < n; i++)
  list.get(i);
```

runs faster than this loop:

```
for (Iterator i=list.iterator(); i.hasNext(); )
  i.next();
```

What I found interesting was that the algorithms always treat the collections as RandomAccess unless they are bigger than some threshold. This makes a lot of sense to me, because I have often found algorithms that we thought to be faster for linked lists actually being slower, as I discussed in Issue 024 of **The Java(tm) Specialists' Newsletter**. The values of the thresholds are of course *not* configurable and look like empirical thumbsuck that would work well with the LinkedList.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

RuntimeException Specifications - Admission of Guilt?

All over the JDK 1.4 java.util.* package, I have seen that the RuntimeExceptions are now also specified in the @throws clause of the JavaDocs. Bruce Eckel and I spent quite a few emails debating the current exception model, and we ended up wondering whether it was not perhaps fundamentally flawed. For example, java.io.IOException has over 30 subclasses that could be the actual exception being thrown. What good is that? When you see that a method throws IOException, what is actually being thrown, and how do you deal with it?

I think that checked exceptions should be scrapped. They don't work. They cause bad code, such as:

```
while(true) {
  try {
    Thread.sleep(1000);
    // do some other work
  } catch (Exception e) {}
}
```

Checked exceptions are responsible for more sloppy code and bugs than any other construct in the Java language. C# does not have checked exceptions, neither does C++. Why does Java have to be encumbered by them?

I think that declaring as comments what runtime exceptions could possibly be thrown by a method is a step in the right direction.

I've discovered some other interesting little snippets, which I will share with you in my next newsletter.

Please remember to send me a quick email if you would like to attend my course in Cape Town to find out how the new JDK 1.4 constructs work. We won't go into this much detail in the course, rather, we'll look at how you can use these new constructs in your work.

Until the next newsletter...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 054] - HashMap requires a better hashCode() - JDK 1.4 Part II

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 54th edition of **The Java(tm) Specialists' Newsletter** sent to 4289 Java Specialists in 85 countries. I seem to have to explain with each newsletter why you have not heard from me for soooo long. Last night I was speaking to a Java guru at Dimension Data in Cape Town, and he mentioned that somehow he had gotten unsubscribed from my newsletter. "No", said I, "you are still subscribed. It has just been rather quiet lately."

We have now finally completed moving into our new home, which includes an office with a view across False Bay all the way to Cape Point (when the weather is nice, which is most of the time :-), so perhaps, the newsletters will appear more frequently. Last week I had to go on an emergency mission to Europe to try help solve some Java deadlock in a very large piece of code. If you get stuck with a deadlock in Java, or even worse, a livelock, you are welcome to contact me and I will see if I can be of help.

Calling all Mac fans & haters! - I am considering buying a Mac notebook for Java development and to develop my courseware. If you have any horror stories (or success stories) regarding Java and Mac OS X, please let me know.

HashMap requires a better hashCode() - JDK 1.4 Part II

My explorations into the JDK 1.4 source code did not come unrewarded. Within a day of starting my expedition, I stumbled across something that ended up baffling several Java experts, including yours truly: **java.util.HashMap now always has a power-of-two number of buckets, irrespective of what size and load factor you specify!!!**

Conventional wisdom says that the buckets in a hash table should be a prime number size. That's what Craig Larman and Rhett Guthrie argued in their excellent (albeit a bit dated) book on Java Performance. The reason why the number of buckets should be prime is so that the hash values are well distributed, even when the hash function is not very well distributed. I'm sure you will find the theory behind it in some dusty CompSci 101 textbook somewhere. The worst possible number of buckets is a power-of-two! So, why on earth did Sun Microsystems deliberately change java.util.HashMap to force it to have a power-of-two number of buckets? (No, "Sun wants to sell more hardware" is not the correct answer ;-)

I posed this question to Bruce Eckel, who also did not know the reason for this. However, he knew Joshua Bloch, the author of java.util.HashMap, so we asked him. Before reading the answer, spend a few minutes thinking about why this could be.

But before I expound the reason, and the problems that came with the change, I want to mention a problem that someone on JavaDesk (an *advanced* yahoogroups

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Java User Group with excellent content) mentioned: Since JDK 1.3, integer division and remainder calculation has become really slow. I've written a basic microbenching framework based on some code originally posted in the JavaDesk group, and also shown in the bug report where the slowdown was described. One has to be careful with Microbenchmarking, as you can read up in this article from the San Francisco conference.

We start with a simple interface

```
/**
 * Benchmark contains some calculation that is executed inside
 * the doCalculation() method a certain number of times.  The
 * number of iterations is returned from the method.
 */
public interface Benchmark {
  int doCalculation();
}
```

Add to it a little performance testing routine:

```
public class PerformanceTest {
  private static final int REPEATS = 10;

  public static void main(String[] args) {
    if (args.length != 1) {
      usage();
    }
    System.out.println(
      "JVM version:" + System.getProperty("java.version"));
    try {
      evaluate((Benchmark)Class.forName(args[0]).newInstance());
      System.out.println();
    } catch (Exception ex) {
      ex.printStackTrace();
      usage();
    }
  }

  private static void usage() {
    System.err.println(
      "usage: java PerformanceTest BenchmarkClass");
    System.err.println(
      "\tBenchmarkClass is a class implementing Benchmark");
    System.exit(1);
  }

  private static void evaluate(Benchmark benchmark) {
    // do the function once to "warm up" HotSpot compiler
    benchmark.doCalculation();
    long average = 0;
    for (int i = 0; i < REPEATS; i++) {
      long time = -System.currentTimeMillis();
      int iterations = benchmark.doCalculation();
      time += System.currentTimeMillis();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      System.out.print(iterations / time);
      System.out.print("  ");
      System.out.flush();
      average += iterations / time;
    }
    System.out.println();
    System.out.println(
      "Average "
       + (average / REPEATS)
       + " iterations per millisecond");
  }
}
```

The most basic test that I want to show is a simple empty loop. This "benchmark" is really quite useless - any half-decent compiler should optimize it away anyway. In addition, if you unroll the loop by 16 steps, you end up doing nothing 16 times and slice the loop into a 16th of the length. The reason for testing this is simply to have some indication of whether my other tests are actually going to measure anything besides the **for** loop. Please also note that *in general* each version of JDK makes programs run *faster*, but that you will always find some benchmark which seems to demonstrate that the old JDK was faster.

```
public class EmptyLoopBenchmark implements Benchmark {
  private static final int ITERATIONS = 10 * 1000 * 1000;
  public int doCalculation() {
    for (int i = 0; i < ITERATIONS; i++) {
      ; // NOP
    }
    return ITERATIONS;
  }
}
```

The results are interesting, although they don't really tell us very much:

```
JVM version:1.2
333333  333333  333333  200000  333333 ...
Average 311056 iterations per millisecond

JVM version:1.3.1_03
250000  200000  250000  166666  250000  ...
Average 217941 iterations per millisecond

JVM version:1.4.0
142857  166666  166666  125000  166666  ...
Average 151785 iterations per millisecond

JVM version:1.4.1-beta
82644  76923  76923  76923  83333  ...
Average 78926 iterations per millisecond
```

Is the JDK 1.4.1 then slower than the JDK 1.2 ?!? Nope, not in general, but as I said, with benchmarks you can find strange things. However, let's look at a benchmark on *addition*. We have to use the val variable after the loop, otherwise a clever compiler could optimize the whole loop away.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public class AdditionBenchmark implements Benchmark {
  private static final int ITERATIONS = 10 * 1000 * 1000;
  private int memory;
  public int doCalculation() {
    int val = 0;
    for (int i = 0; i < ITERATIONS; i++) {
      val = val + i;
    }
    memory = val;
    return ITERATIONS;
  }
}
```

The results on my little Asus Notebook are as follows:

```
JVM version:1.2
333333 333333 200000 333333 500000 ...
Average 344999 iterations per millisecond

JVM version:1.3.1_03
200000 200000 200000 243902 200000 ...
Average 209390 iterations per millisecond

JVM version:1.4.0
125000 125000 125000 123456 125000 ...
Average 123853 iterations per millisecond

JVM version:1.4.1-beta
83333 76923 76335 90909 76923 ...
Average 79486 iterations per millisecond
```

These results at best demonstrate that addition is blindingly fast, as we would expect. Note that we have to compare the addition benchmark to the empty loop benchmark to realise that the actual addition operation is negligible. However, what about remainder, the % operator?

```
public class RemainderBenchmark implements Benchmark {
  private static final int ITERATIONS = 10 * 1000 * 1000;
  private int memory;
  public int doCalculation() {
    int val = 0;
    for (int i = 0; i < ITERATIONS; i++) {
      val = i % 11;
    }
    memory = val;
    return ITERATIONS;
  }
}
```

Here we see that performance has dropped significantly since JDK 1.2, as described in the bug report mentioned earlier:

```
JVM version:1.2
62500 62500 62111 58823 66666 ...
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Average 62520 iterations per millisecond

JVM version:1.3.1_03
17513 17513 17513 17211 17513 ...
Average 17457 iterations per millisecond

JVM version:1.4.0
14903 16920 16638 15360 16366 ...
Average 16051 iterations per millisecond

JVM version:1.4.1-beta
17211 17513 17211 16920 16920 ...
Average 17217 iterations per millisecond

These results is somewhat disturbing! Isn't that one of the important functions that gets executed in the traditional java.util.HashMap class? The engineers at Sun (Joshua Bloch and Doug Lea) decided to rather use a bit masking approach instead of remainder, because that is much faster:

```
public class MaskingBenchmark implements Benchmark {
  private static final int ITERATIONS = 10 * 1000 * 1000;
  private int memory;
  public int doCalculation() {
    int val = 0;
    for (int i = 0; i < ITERATIONS; i++) {
      val = i & 0x000000ff;
    }
    memory = val;
    return ITERATIONS;
  }
}
```

You can see that the performance values are far more encouraging:

JVM version:1.2
166666 142857 166666 166666 125000 ...
Average 158416 iterations per millisecond

JVM version:1.3.1_03
142857 142857 100000 140845 111111 ...
Average 131624 iterations per millisecond

JVM version:1.4.0
111111 142857 125000 125000 123456 ...
Average 128813 iterations per millisecond

JVM version:1.4.1-beta
76923 76923 83333 76335 83333 ...
Average 80849 iterations per millisecond

In order to protect Sun against claims by the intellectual proletariat ;-), who have up to now managed to get away with writing bad hash functions, Josh wrote a super-fast rehash() function inside java.util.HashMap that attempted to redistribute the bits a little bit. Good hash code writers were rewarded with a nice improvement in

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

performance for java.util.HashMap.get() in **JDK 1.4.0** thanks to the new masking trick:

```
import java.util.*;

/**
 * The lower-order bits are used as part of this hashcode
 * which is good if you are using JDK 1.4.0
 */
public class GoodHashcodeBenchmark implements Benchmark {
  private static final int ITERATIONS = 1000 * 1000;
  private Object memory;
  private HashMap map = new HashMap(203);
  private Integer[] values;
  public GoodHashcodeBenchmark() {
    for (int i=0; i < 1000; i++) {
      map.put(new Integer(i), "Value");
    }
    TreeSet keys = new TreeSet(map.keySet());
    values = (Integer[])keys.toArray(new Integer[1000]);
  }
  public int doCalculation() {
    for (int i = 0; i < ITERATIONS; i++) {
      memory = map.get(values[i%1000]);
    }
    return ITERATIONS;
  }
}
```

When we run this benchmark, we get the following values:

```
JVM version:1.2
5555 5882 5882 5524 5555 5555 5555 5847 5555 6250
Average 5716 iterations per millisecond

JVM version:1.3.1_03
3846 3690 3846 3436 3846 3571 3831 3703 3690 3571
Average 3703 iterations per millisecond

JVM version:1.4.0
6250 6250 5847 6250 5882 5882 6622 5882 5882 5882
Average 6062 iterations per millisecond

JVM version:1.4.1-beta
4149 4347 4347 4329 4000 4329 4347 4545 4149 4347
Average 4288 iterations per millisecond
```

Despite the slowdown of JDK 1.[34].x with regards to remainder calculations, Joshua Bloch and Doug Lea still managed to actually make HashMap.get() function in the JDK 1.4.0 run faster than in the JDK 1.2! The observant reader would have noticed that JDK 1.4.1 runs slower than JDK 1.4.0 but faster than JDK 1.3.1. Before I explain why, let's have a look at what happens when you write a bad hash function:

```
import java.util.*;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
/**
 * The lower-order bits are NOT used as part of this hashcode
 * which is bad if you are using JDK 1.4.0
 */
public class BadHashcodeBenchmark implements Benchmark {
  private static final int ITERATIONS = 1000 * 1000;
  private Object memory;
  private HashMap map = new HashMap(203);
  private Integer[] values;
  public BadHashcodeBenchmark() {
    for (int i=0; i < 1000; i++) {
      map.put(new Integer(i * 1024), "Value");
    }
    TreeSet keys = new TreeSet(map.keySet());
    values = (Integer[])keys.toArray(new Integer[1000]);
  }
  public int doCalculation() {
    for (int i = 0; i < ITERATIONS; i++) {
      memory = map.get(values[i% 1000]);
    }
    return ITERATIONS;
  }
}
```

The result is obvious. Since the new JDK 1.4.0 HashMap is ultra-sensitive to lower-order bits being the same, due to it using masking instead of remainder, the HashMap basically becomes a linked list and the lookup for a single entry becomes O(n) instead of O(1). Oops:

JVM version:1.2
5524 5555 5882 5882 5524 5882 5555 5555 5524 5555
Average 5643 iterations per millisecond

JVM version:1.3.1_03
3571 3690 3571 3703 3558 3703 3558 3703 3571 3690
Average 3631 iterations per millisecond

JVM version:1.4.0
173 172 154 171 173 171 168 167 172 172
Average 169 iterations per millisecond

JVM version:1.4.1-beta
4149 4347 4347 4149 4166 4347 4329 4347 4347 4329
Average 4285 iterations per millisecond

Bug reports started surfacing on the bug parade saying that the HashMap in JDK 1.4.0 had a bug that caused it to be very slow. The following is an excerpt from a three-way conversation between Joshua Bloch, author of java.util.HashMap, Bruce Eckel and myself (quoted with permission):

Joshua Bloch: *The downside of using a power-of-two is that the resulting hash table is very sensitive to the quality of the hash function (hashCode). It is imperative that any change in the input must affect the low order bits of the hash value.*

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

*(Ideally, it should affect all bits of the hash value with equal likelihood.) Because we have no assurance that this is true, we put in a secondary (or "defensive") hash function when we switched to the power-of-two hash table. This hash function is applied to the results of hashCode before masking off the low order bits. Its job is to scatter the information over all the bits, and in particular, into the low order bits. Of course it has to run *very* fast, or you lose the benefit of switching to the power-of-two-sized table. The original secondary hash function in 1.4 turned out to be insufficient. We knew that this was a theoretical possibility, but we thought that it didn't affect any practical data sets. We were wrong. The replacement secondary hash function (which I developed with the aid of a computer) has strong statistical properties that pretty much guarantee good bucket distribution.*

The rehash() method in JDK 1.4.1 is quite interesting, but let's hope that it does a good enough job. From my tests, it seems to distribute the bits beautifully, but you will have to test (I don't know how!) that your hash function is compatible with the rehash method of HashMap. Obviously, having a more complicated rehash() method causes a drop in performance (when the hash function was good anyway), but it gives us a better average performance than JDK 1.4.0 and JDK 1.3.1.

Moral of the story - write your own!

If you need to squeeze the last ounce of performance out of the JDK and you need to do that with hash tables, you should probably write your own stripped-down version of HashMap that works specifically for your key-domain.

Conclusion

I now understand the reason for using power-of-two for HashMap buckets, but, I am also disappointed. The old mechanism was so much more predictable in terms of distribution of keys into buckets. How am I going to explain this new mechanism to Java beginners?

On the other hand, this new knowledge allows me (and you) to amble up to people who claim to know Java and say: "Did you know that HashMap in JDK 1.4 forces the number of buckets to be a power-of-two?" If the innocent victim responds: "Yes", you can follow-up your question with "Why?" Try it out - and let me know the response :-)))

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 054b] - Follow-up to JDK 1.4 HashMap hashCode() mystery

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

This is just a quick follow-up to the newsletter sent this morning at 2:00am South African time. It seems I was not very careful with the source code and a few errors crept in. Please have a look at the archive for a corrected edition. After finally collapsing in bed at 2:15am, I was rudely reminded that I have a daughter of 11 months old. She kept up her antics until 6:00am, when I had to get up to carry on presenting my Java course. *sigh* - today was not a good day!

Joshua Bloch wrote to me after last night's newsletter, sending me some more information about the remainder performance mystery, and I feel I should pass the information on to you, my readers:

Joshua Bloch: *By the way, I now know more about what's going on with mod/division. There is a collection of techniques for doing fast division by a constant. These techniques are covered in great detail in Chapter 10 of a marvelous new book with the unlikely title of "Hacker's Delight" by Henry Warren. It turns out that the old ("Classic") VM knew some of these tricks, but Hotspot, in releases up to 1.4, did not. While 1.4.1 can do some of this stuff, I suspect that later releases will do more.*

I wrote a FairRemainderBenchmark that calculates the remainder with a variable, the way that the old HashMap would have done, and alas, the speed of the various JDKs is roughly the same:

```
import java.util.Random;
public class FairRemainderBenchmark implements Benchmark {
  private static final int ITERATIONS = 10 * 1000 * 1000;
  private int memory;
  public int doCalculation() {
    int val = 0;
    Random rand = new Random(0);
    int bucket_size = (int)(rand.nextDouble() * 101) + 1;
    for (int i = 0; i < ITERATIONS; i++) {
      val = i % bucket_size;
    }
    memory = val;
    return ITERATIONS;
  }
}
```

The performance is now quite similar between the old and the new versions of the JVM:

```
JVM version:1.2
18867 18832 18484 19193 18832 18518 19193 18832 18484 18832
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Average 18806 iterations per millisecond

JVM version:1.3.1_03
18867 18148 18832 18484 18484 18867 18832 18832 18148 18484
Average 18597 iterations per millisecond

JVM version:1.4.0
18832 19193 19230 19193 19193 19193 19193 19230 19193 19193
Average 19164 iterations per millisecond

JVM version:1.4.1-beta
17825 18148 18148 18148 17543 18484 18148 18148 17513 18518
Average 18062 iterations per millisecond

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 055] - Once upon an Oak ...

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 55th edition of **The Java(tm) Specialists' Newsletter** sent to 4461 Java Specialists in 85 countries. We had a good growth in the last two weeks, thanks to your concerted effort in forwarding this newsletter to your friends. There are three people in particular that I want to thank, although many more would be worthy of a mention:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Mike Cannon-Brookes mentioned me on his rebelutionary weblog. Mike works for Atlassian, the company that produced the bug tracking tool JIRA.
Henri Yandell mentioned me on his bayard weblog.
Kevin Oliver from Salesforce.com for convincing many of his colleagues to subscribe.

Please see the end of this email for information about my upcoming Design Patterns Course.

When I studied for Java Certification a few years ago, I used a little program to help me along, called JCertify, published by EnterpriseDeveloper.com. I really enjoyed this little program, and I hope to think that my error reports mean that the questions now have correct answers ;-) Due to my long-standing relationship with the original author (who is also subscribed to **The Java(tm) Specialists' Newsletter**), they have given our newsletter a special discount link, where you can get a 20% discount until the 7th of September. I don't benefit in any way if you purchase JCertify - I am simply passing along information about a great little program that you can benefit from.

Unce upon an Oak ...

A few weeks ago I was talking to someone about the origins of Java, and it occurred to me that I had big gaps in that part of world history. History had never been my forte, at school this was my weakest subject at 35% (at my worst level). Perhaps I should add that I was very weak with *South African history*, which at the time of me going to high school supposedly started with the white settlers arriving on the shores, and consisted almost entirely of remembering dates. I found such parrotting frivolous, and so my marks were not too hot.

Trying to fill my gaps of Java's history, I started digging around on Sun's website, and eventually stumbled across the Oak Language Specification for Oak version 0.2. Oak was the original name of what is now commonly known as Java, and this manual is the oldest manual available for Oak (i.e. Java). For more history on the origins of Java, please have a look at The Green Project and Java(TM) Technology: An Early History. I printed the manual and kept it next to my bed in case of insomnia (so I thought). When I started reading it, I discovered answers to ancient questions: Why can we have only one public class per .java file? Why are protected members also accessible from the same package? Why do **short** fields use as much memory as **int** fields (at least pre-JDK 1.4)?

In this newsletter I want to highlight the differences between Java 0.2 (i.e. Oak 0.2) and what we have now. For your reference, I will include the section numbers in the Oak 0.2 specification.

Why is each public class in a separate file? (Section 1)

This is a question that I have frequently been asked during my courses. Up to now I have not had a good answer to this question. In section 1, we read: "Although each Oak compilation unit can contain multiple classes or interfaces, at most one class or interface per compilation unit can be public".

In the sidebar it explains why: "This restriction is not yet enforced by the compiler, although it's necessary for efficient package importation"

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

It's pretty obvious - like most things are once you know the design reasons - the compiler would have to make an additional pass through all the compilation units (.java files) to figure out what classes were where, and that would make the compilation even slower.

One-line Javadoc comments (Section 2.1)

Oak had the ability to write a one-line JavaDoc comment.

```
public class OneLineJavaDoc {
  //* The number of lines allowed in a document.
  public int numberOfLines;
}
```

This is *fortunately* not supported in Java as it is quite confusing.

Additional Keywords (Section 2.3)

Oak had some additional keywords that are not currently used in Java:

`ushort`, `string`, `Cstring` and `unsynchronized` were already obsolete in version 0.2 of Oak. Isn't it amazing how quickly things become obsolete in our industry?
`clone`, `const` and `goto` were keywords
`protect` and `unprotect` were nasty keywords for writing terrible exception code. More about that later...
`enum` was a keyword, but in the sidebar it said: `enum` isn't implemented yet. So, in answer to the question: Why does Java not have enum? The answer is not some heavy object-oriented philosophical answer about polymorphism and strategy patterns. It is simply that James Gosling didn't implement it in time and his management forced him to push the language out of the door before he could finish the feature :-) Unsigned integer values (Section 3.1)

The specification says: "The four integer types of widths of 8, 16, 32 and 64 bits, and are signed unless prefixed by the unsigned modifier.

In the sidebar it says: "unsigned isn't implemented yet; it might never be." How right you were.

Storage of Integer values (Section 3.1)

In my newsletter on Determining Memory Usage in Java, I noted that when you have a data member of byte or short, that they still use up at least 4 bytes of memory. This was changed in JDK 1.4, but the historical reason is found in the Oak spec:

"A variable's type does not directly affect its storage allocation. Type only determines the variable's properties and legal range of values. If a value is assigned to a variable that is outside the legal range of the variable, the value is reduced modulo the range."

I wish I had known that when I wrote my first Java program. I spent a lot of time deciding which data types I wanted to use in order to save memory, but I was

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

actually wasting my time. Please note that this has changed as of JDK 1.4, so now it does help to use bytes and shorts to reduce the memory footprint.

Arrays (Section 3.5)

In Oak, you were able to declare arrays as follows:

```
int a[10];
a[5] = 1;
```

However, we were also able to declare an array in the current Java fashion, with **new**. I think having two ways of doing the same thing causes confusion, so I am very pleased that this way of making new arrays has been removed.

Const vs. Final (Sections 4.6 and 4.9)

It appears that final was initially only meant to be used for classes and methods, and that const was used for making fields constant.

Private did not exist (Sections 4.10)

This is the most surprising part of Oak 0.2. There were only three access levels as opposed to our current four. There was **private** which did not require a keyword, and which equated to our current "package private" or "friendly" or "package access" type of access level. All classes in a particular package could use all variables and methods declared in the classes in that package, regardless of **public**, **protected** and **private** declarations. I am very glad that they introduced a more private version of **private**, one that was only accessible within the class.

The lack of private as we know it today explains why when a member is protected, it is also accessible from within the same package. When Oak was written, protected was obviously accessible within the package because their private (our "package access") was the most restrictive access level. I seem to remember that in JDK 1.0 we had a fifth access level called "private protected", which meant that *only* subclasses could access the member.

This piece of surprising history also explains why fields are not private by default - they actually were "private" originally when private meant different things.

I don't need to emphasize how pleased I am that we now have the more restrictive "private" modifier. Without that, our industry would be in even more trouble.

Abstract Methods (Section 5)

Abstract methods were defined as in C++:

```
public interface Storing {
  void freezeDry(Stream s) = 0;
}
```

Assertions and Pre/Postconditions (Sections 7, 7.1 and 7.2)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Unfortunately the assertions in Oak 0.2 were not implemented in time, so they were thrown out to satisfy the release deadline. In case you think that assertions are back in JDK 1.4, have a look at the power that "those" assertions gave you:

```
class Calender {
  static int lastDay[12]=
    {31,29,31,30,31,30,31,31,30,31,30,31};
  int month assert(month>=1 && month<=12);
  int date assert(date>=1 && date<=lastDay[month]);
}
```

While objects are not required to obey the legality constraints within methods, the constraints are enforced at the entry and exit of every public and protected method.

I wish that James Gosling had worked a few extra weekends (if that were possible) to finish the implementation of **assert** as it appeared in the original Oak spec. Preconditions and Postconditions were also loosely defined, but were also kicked out due to time pressure. Pity.

Post-incrementing Strings (Section 8.1.4)

In Oak, you were able to post-increment a String. You could literally say s1++;, which was equivalent to s1 = s1 + 1;. The post-increment statement is often (if not always) implemented as +=1 so it seems that this also was true for Strings. Fortunately this is not allowed in Java anymore.

Goto ... (Section 9.3)

Of course, being based on C, Oak included the infamous **goto** statement. Fortunately this is not in Java.

Exceptions (Section 9.4)

Where does the name RuntimeException come from? Aren't all exceptions thrown at runtime? These are exceptions which are thrown *by the runtime system*.

In Oak 0.2, all exceptions were unchecked, meaning that there was no **throws** clause. My guess is that checked exceptions were only added once the whole exception hierarchy had already been set in wet concrete. I would have had a separate branch for checked exceptions, something like:

```
public class Throwable { }

/** Serious errors in the virtual machine */
public class Error extends Throwable { }

public class CheckedException extends Throwable { }
public class IOException extends CheckedException { }

public class UncheckedException extends Throwable { }
public class NullPointerException extends UncheckedException { }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

This way you would avoid catching unchecked exceptions when you catch CheckedException. However, as it appears, the exception class hierarchy was developed before the idea of checked vs. unchecked exceptions, so we are stuck with an exception mechanism that will cause headaches for generations to come.

Asynchronous Exceptions (Section 9.4.2)

If your program gets an asynchronous exception, you are dead. A few weeks ago I was looking at a program that was throwing OutOfMemoryError. This can happen at any time really, which is why it is called an *asynchronous* exception. Another place where this can happen is with Thread.stop(). As you can imagine, this is inherently dangerous, that is why it is deprecated. In Oak, life was even more dangerous. You could cause an asynchronous exception in another thread using Thread's postException() method. Now *that* was dangerous! Imagine if other threads could cause asynchronous exceptions at any place in your code!

In order to safeguard your code, you could "protect" it. If you wanted to indicate that you had some critical code which could not handle asynchronous exceptions, you did the following:

```
protect {
  /* critical section goes here */
}
```

And code that was quite happy with asynchronous exceptions did the following:

```
unprotect {
  /* code that can afford asynchronous exceptions */
}
```

There was a note in the sidebar saying that the default would probably be changed to *not* allow asynchronous exceptions except in explicitely *unprotected* sections of code.

I'm very glad that this feature was binned. I cannot imagine how complex our Java programs would have become with it in place.

Summary

That's it. The rest of the manual was filled with a Glossary and an Index, to push the manual to 38 pages.

Even though this manual was written way back in 1994, it provided for fascinating reading, even late at night ;-)

Until next time ...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 056] - Shutting down threads cleanly

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 56th edition of **The Java(tm) Specialists' Newsletter** sent to 4609 Java Specialists in 85 countries. Whenever I think I wrote a "killer" newsletter, I get a lukewarm response from my readers, and when I think I wrote a flop, I get accolades, like with my last newsletter about Oak. I expected rotten tomatoes in the form of "here are 29 other newsletters about Oak and modern Java". Perhaps I should *try* and write flops, then I will have more successes ;-)

When I was in Germany from June to August 2000, we lived in a village about 45 minutes by train from Infor AG. Every day I had 1.5 hours to read books about various topics. Over the course of 3 months, I got through a lot of reading, I even read The Java Virtual Machine Specification! Much like the Oak specification, it is a *must* for any serious Java developer to read. The more I read in that spec, the more I realised that the hope of "write once, run anywhere" was a wishful dream. Another book that I devoured is Concurrent Programming in Java, 2nd Ed by Doug Lea, one of the writers of java.util.HashMap. My thoughts on shutting down threads is based on ideas I gleaned from those two books, so before you shoot me down, please read the books. The Java 2 Performance and Idiom Guide is also an excellent book to read for the reason that it got me thinking about performance the way that no other book has. It is quite dated, which cannot be avoided with a book about a language that changes so quickly. That book also agrees with my approach of shutting down threads.

My **September Design Patterns Course** starting tomorrow is completely full. If you missed the September course, please see the end of this email for information about our upcoming **November Design Patterns Course**. That course is mainly aimed at South Africans, due to the difficulty of getting into this country - a visa from Germany can take three weeks to get approved! We do not want you to know how fantastic it is here, so we make it as difficult as possible for you to get here, and especially to *stay* here. If you thought getting into the USA was hard, try South Africa.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

All is not lost, however. If you live in a country far far away from South Africa and your company would benefit from a solid Design Patterns Course (which company would *not*?!?), I will gladly give you a quote for **me** to come to **you** and present the course at your company. Simply send me an email and you will hear from me personally within 24 hours.

How to shutdown threads cleanly

I remember starting off with JDK 1.0, and playing with Threads. I would start a Thread, and then to stop it, I simply called stop(). It was the most obvious thing to do, with disastrous consequences. If we refer back to the Oak Newsletter, that was the same as having *all* your code unprotected, and receiving an **asynchronous** exception in your Thread. The exception that is thrown asynchronously is java.lang.ThreadDeath. Needless to say, the more I think about it, the more I am baffled that they allowed that in the first place, especially in the absence of the protect keyword that they had in Oak. Using stop() is incredibly dangerous, as it will kill your thread even if it is in the middle of something important. There is no way to protect yourself, so if you spot code that uses stop(), you should frown.

So, how do you shutdown a thread cleanly? The developers of Java have actually left the protect mechanism in place for us to use, it is just named differently. First of all, we should never use stop(). Ever. Period. Do not even think of using it. Sun should remove it from java.lang.Thread as soon as possible. Don't. No. Noooooo. Doooown. Secondly, the only place where we are allowed to receive an exception to tell us that the thread is being shutdown is while the thread is blocked. Getting a shutdown notification at *any* other time would be dangerous and nondeterministic. The way this works is via the java.lang.InterruptedException. I admit that InterruptedException is not the most obvious choice of *name* for indicating that another thread is trying to shutdown your thread.

I have seen code that uses a boolean flag to indicate whether the thread is running or not. However, Java already provides that flag in the form of the interrupted flag, so why duplicate effort? Usually, the code would work something like this:

```
public class UsingFlagToShutdownThread extends Thread {
  private boolean running = true;
  public void run() {
    while (running) {
      System.out.print(".");
      System.out.flush();
      try {
        Thread.sleep(1000);
      } catch (InterruptedException ex) {}
    }
    System.out.println("Shutting down thread");
  }
  public void shutdown() {
    running = false;
  }
  public static void main(String[] args)
      throws InterruptedException {
    UsingFlagToShutdownThread t = new UsingFlagToShutdownThread();
    t.start();
    Thread.sleep(5000);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    t.shutdown();
  }
}
```

What is so bad with that code? This example is not too bad, since the longest we would wait unnecessarily would be one second. However if we normally sleep for 30 seconds, then it could take a while before your program is completely shut down. This is especially true if you have a lot of threads and you join() each one to make sure that it does finish.

Java has another mechanism that you should rather use: simply interrupt the thread. The code would then look like this:

```
public class UsingInterruptToShutdownThread extends Thread {
  public void run() {
    while (true) {
      System.out.print(".");
      System.out.flush();
      try {
        Thread.sleep(1000);
      } catch (InterruptedException ex) {
        Thread.currentThread().interrupt(); // very important
        break;
      }
    }
    System.out.println("Shutting down thread");
  }
  public static void main(String[] args)
      throws InterruptedException {
    Thread t = new UsingInterruptToShutdownThread();
    t.start();
    Thread.sleep(5000);
    t.interrupt();
  }
}
```

I must admit that I have not seen many programmers handle InterruptedExceptions correctly, i.e. using my way ;-) Most of the time, programmers view InterruptedException as an irritating checked exception that they have to catch, but which they usually ignore:

```
while (true) {
  // ... do something
  try {
    Thread.sleep(30000);
  } catch (InterruptedException ex) {}
}
```

Why do we have to interrupt the thread again?

In my example, after I caught the InterruptedException, I used Thread.currentThread().interrupt() to immediately interrupted the thread again. Why is this necessary? When the exception is thrown, the interrupted flag is cleared, so if

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

you have nested loops, you will cause trouble in the outer loops. Consider the following code:

```
public class NestedLoops extends Thread {
  private static boolean correct = true;
  public void run() {
    while (true) {
      System.out.print(".");
      System.out.flush();
      for (int i = 0; i < 10; i++) {
        System.out.print("#");
        System.out.flush();
        try {
          Thread.sleep(100);
        } catch (InterruptedException ex) {
          if (correct) Thread.currentThre ad().interrupt();
          System.out.println();
          System.out.println("Shut down inner loop");
          break;
        }
      }
      try {
        Thread.sleep(1000);
      } catch (InterruptedException ex) {
        if (correct) Thread.currentThread().interrupt();
        System.out.println();
        System.out.println("Shut down outer loop");
        break;
      }
    }
    System.out.println("Shutting down thread");
  }
  private static void test() throws InterruptedException {
    Thread t = new NestedLoops();
    t.start();
    Thread.sleep(6500);
    t.interrupt();
    t.join();
    System.out.println("Shutdown the thread correctly");
  }
  public static void main(String[] args)
      throws InterruptedException {
    test();
    correct = false;
    test();
  }
}
```

When you run this code, you will see something like this:

```
.##########.##########.##########.######
Shut down inner loop

Shut down outer loop
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Shutting down thread
Shutdown the thread correctly
.##########.##########.##########.######
Shut down inner loop
.##########.##########.##########.##########.##########.
etc.

Herein lies the danger with this approach: if some library incorrectly handles InterruptedException then your code will not shut down correctly.

From a purely theoretical view, you should use the interrupt mechanism of threads to shut them down. However, you have to be very careful that you use that mechanism throughout your code, otherwise you will not be able to shut down all your threads.

What about threads blocked on IO?

Threads can be blocked on wait(), sleep(), waiting to enter a synchronized block or waiting on some IO to complete. We cannot shut down a thread waiting to enter a synchronized block, so if you have a livelock or deadlock you will not be able to shut down your system cleanly. wait() and sleep() both throw an InterruptedException, as does join(). But, what about when you're blocked on IO? There is an exception called java.io.InterruptedIOException, which is supposed to cover the situation where you interrupt a thread that is waiting on some IO to complete. As you might have guessed, it is not implemented consistently. It works for piped streams, but none of the others seem to have that effect.

If you want to stop a thread waiting on a socket, you will have to unfortunately close the socket underneath the thread. Fortunately, the interrupt() method is not final, so you can override it to also close the socket. Inside the **catch** clause of java.io.IOException you can then check whether the thread has been interrupted or not:

```
import java.io.IOException;
import java.io.InputStream;
import java.io.InterruptedIOException;

public class BlockedOnIO extends Thread {
  private final InputStream in;
  public BlockedOnIO(InputStream in) {
    this.in = in;
  }
  public void interrupt() {
    super.interrupt();
    try {
      in.close();
    } catch (IOException e) {} // quietly close
  }
  public void run() {
    try {
      System.out.println("Reading from input stream");
      in.read();
      System.out.println("Finished reading");
    } catch (InterruptedIOException e) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      Thread.currentThread().interrupt();
      System.out.println("Interrupted via InterruptedIOException");
    } catch (IOException e) {
      if (!isInterrupted()) {
        e.printStackTrace();
      } else {
        System.out.println("Interrupted");
      }
    }
    System.out.println("Shutting down thread");
  }
}
```

For shutting down threads reading from sockets, we would do something like this:

```
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class BlockedOnSocketIO {
  public static void main(String[] args)
      throws IOException, InterruptedException {
    ServerSocket ss = new ServerSocket(4444);
    Socket socket = new Socket("localhost", 4444);
    System.out.println("Made socket, now reading from socket");
    Thread t = new BlockedOnIO(socket.getInputStream());
    t.start();
    Thread.sleep(5000);
    t.interrupt();
  }
}
```

When we run our code, we see the following:

```
Made socket, now reading from socket
Reading from input stream
Interrupted
Shutting down thread
```

Alternatively, when we use Pipes:

```
import java.io.IOException;
import java.io.PipedInputStream;
import java.io.PipedOutputStream;

public class BlockedOnPipedIO {
  public static void main(String[] args)
      throws IOException, InterruptedException {
    PipedInputStream in =
      new PipedInputStream(new PipedOutputStream());
    Thread t = new BlockedOnIO(in);
    t.start();
    Thread.sleep(5000);
    t.interrupt();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
  }
}
```

When we run that code, we see the following:

```
Reading from input stream
Interrupted via InterruptedIOException
Shutting down thread
```

Unfortunately, the IO library in Java is not consistent, so you have to cater for both possibilities in your shutdown methods.

I hope that this newsletter will be as useful to you as it has been to me. Shutting down threads cleanly is unfortunately not as easy as it should be, but the mechanism in this newsletter is superior to calling stop() (and thereby using an **asynchronous exception**) and it is also better than using a flag to indicate whether the thread is supposed to carry on running or not.

The only problem with my approach is that if you use some library that does not handle InterruptedException correctly, you will have problems shutting down your thread. You might have to have a separate thread that calls join() with a timeout and repeatedly interrupts the thread until it is shut down.

That's the end of the newsletter. The birds are singing to celebrate spring, my baby sister is here to visit, so we are now going to celebrate life with a Cuban cigar :-)

Heinz

P.S. Please remember to contact me by sending a short email if you are interested in a quote for an inhouse Design Patterns Course.

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 057] - A Tribute to my Dad, Hans Rudolf Kabutz

Author: Dr. Heinz M. Kabutz

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 57th edition of **The Java(tm) Specialists' Newsletter** sent to 4686 Java Specialists in 86 countries.

This is not one of my usual newsletters, i.e. you won't find anything about Java in here. However, the content of what I am about to say has everything to do with the birth and existence of **The Java(tm) Specialists' Newsletter**.

On the morning of Wednesday the 18th of September 2002, my greatest inspirator passed away suddenly at the age of 64 from heart failure. No one will ever know what my father meant to me. He was constantly sending me emails, phoning me, encouraging me to try new ideas, to publish these newsletters, to go where others did not dare to go. If it were not for him, I would not even have started writing these newsletters. I certainly would not have published more than 50 in the period of two years and reached 86 countries without his constant support.

Whenever a new country appeared on my list (latest is the Democratic Republic of Congo), I would phone my dad to tell him, and he would laugh and say: "Mach weiter so mein Sohn" (Keep on going, my son). It was my dad who first noticed the newsletters that Peter Carruthers was writing, and forwarded them to me. Peter's newsletters became the basis on which I have worked for this newsletter. That is why I can say that if it were not for my father, **The Java(tm) Specialists' Newsletter** would never have been started.

I remember telling him when I reached the 400 subscriber mark, and him laughing and being so happy. When I told him that I was going to Mauritius to present courses there, he laughed and laughed.

I would not have pushed through and gotten my PhD in Computer Science without his help. The fact that my father had produced a "doctor" was a constant source of amusement for him. "Mein kleiner Heinz", he would say. Every facet of my life has been influenced and moulded by my father. Even going to university was almost entirely his fault. He wanted me to study accounting, but when I refused and said I wanted to do Computer Science instead, he accepted it and supported me all the way.

My father was a giant of a man, about 1.94m tall, quite intimidating. The person that he looked like most was Bud Spencer, just not quite as covered. He used his size and presence to get his way with people. However, inside, my father was very sensitive, very caring, always helpful. As I spent last Wednesday phoning his old friends and telling them the sad news, they would relate to me stories of how my dad had helped them when their car broke down in the middle of the night, or how he had given them sound business advice in a tough financial situation. If ever you had a problem, all you had to do was phone Hans, and he would come and help you. It did not matter when or what.

During my childhood, my father worked constantly, twelve hours a day at his factory, and then several hours doing administration at home. We hardly saw him

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

due to his heavy schedule. My teenage years were extremely tense. My father was unbelievably stubborn. Almost as stubborn as me. On top of that, I cannot tell a lie. The combination of stubbornness and honesty meant that instead of sneaking out, I would tell my father when I was going out, and we would have a big dogfight about when I would be back and where I was going. Once we had such a big confrontation that he locked the PCs keyboard away for about two weeks. After a while his machine stopped working (guess why ;-) and the keyboard reappeared. I think of all the children (I have two brothers and one sister), I was the most punished. It didn't help. I am still as stubborn as I have always been ;-)

During the last few years of my father's life we became really close. He would share information about his business with me that no one else in the world knew about, not even my mother. I would tell him the minutest details of what I was doing and he would laugh and give me even better ideas of all I could do. In May this year, I phoned my dad to speak to him and he had just been to his doctor with Arrhythmia (an abnormal heartbeat rhythm). He told me in a weak voice "I am getting old" and then the phone went dead. I had a terrible shock, but it was just his battery that had run out of juice. Shortly after that, we had a family meeting to discuss what we would have to do in the (unlikely) event of his death. We all agreed that my father would still live for another twenty years, but nevertheless it would be good to be prepared.

My father believed in laughter, but not in death. He was never scared of dying, because he had a strong relationship with Jesus. He has been faithfully serving in his church the Stadtmission Kapstadt, for the last 26 years. For the last few years he has been the chief elder at the church. He is so much part of that church that they cannot even imagine what it will be like without him. It was my father's inspiration that caused our whole family to believe in Jesus. He would be very cross if he were to see me mourning for him - he would tell me to "not be ridiculous" since he is in a better place now. He would probably make a joke about "I can see all the way to Malmesbury from up here". To my father, death was just part of life. He would bellow a laugh "Ha Ha Ha" about anything.

I want to thank my father for not leaving us in the lurch when times were really difficult. One time on Christmas Eve, the Sheriff of the Court came to our house to start confiscating our property due to my father's indebtedness. My father parked his Combi around the block and walked home when he saw the Sheriff's car outside. Then boxes and boxes of fruit and vegetables arrived at our house from an anonymous donour at my parent's church. When we were kids, money was **always** tight, even though we lived in a magnificent home in one of the most exclusive suburbs of Cape Town. One thing I learnt from that is that people who live in fancy houses do not necessarily have a lot of spending money, infact it is usually the opposite.

Most of my holidays were spent slaving away in my father's drinking straw factory (someone **has** to make them). For many years, my father rented the most shabby facilities in Woodstock (a really crummy area of Cape Town, full of factories and drug dealers). Once we arrived in the morning to discover a vagrant lying outside, stabbed to death. After about twenty years in Woodstock, my father built a new factory building in a new industrial development.

My father used to joke that you can neither live nor die from making such a petty consumable. I went on three business trips with my dad when I was a teenager, driving all around the country trying to get more business. It was always interesting

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

watching my father relate to his customers. My father also used to joke that his business is "clutching at straws", until he became the best straw manufacturer in Africa, then his phrase became "clutching at straws is my business". He managed to build up his entire business without ever bribing anybody, which in Africa is quite an achievement.

I will always fondly remember my father, Hans Rudolf Kabutz, who always ended his long, loborious, detailed-to-the-point emails with: "Liebe Gruesse sendet Vater HansR". He was a fantastic husband to my mother Sigrun Ingeborg Kabutz nee Kuhlmann, and the best father ever to Rudolf Thomas Kabutz, Marten Herbert Kabutz, Bettina Sigrun Kabutz and me, Heinz Max Kabutz. His three daughters-in-law Helene Kabutz (my wife) nee Kytides, Karin Kabutz nee Ebrecht and Alex Kabutz nee Glatz, and his six grandchildren Maximilian, Constance, Heiko, Thomas, Meike and Kyra. We will all miss you terribly.

To the most eccentric man, who stubbornly insisted all his life that he would not get much older than 62, and who stubbornly went to the Lord "according to plan": We love you and we will sorely miss you.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 058] - Counting bytes on Sockets

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 58th edition of **The Java(tm) Specialists' Newsletter** sent to 4814 Java Specialists in 86 countries.

I was quite blown away by your support and help offered after last week's newsletter. All I can say is "thanks". Thank you especially for the encouragement to "carry on" with the newsletters, obviously that was my intention :-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

My mother, my sister and I are now running my father's company, manufacturing drinking straws. For my sister and me it is a part-time occupation, since we both have other occupations. However, we have all been involved with the business for the last 20 years, so fortunately we know what is involved. Don't expect to see less newsletter though - they are still my #1 priority :-) and I will fit them in inbetween writing Java code, training people on Design Patterns and Java, and having staff meetings. Talking of Design Patterns, I have another course running from the 12th to the 14th of November, see bottom of newsletter. I am also planning a Java 2 Standard Edition Course before the end of the year, so if you would like to attend, **please pop me an email** so that we can arrange a time that suits everybody.

Counting bytes on Sockets

Background

At the end of 2001, William Grosso, author of Java RMI, offered to send me a copy of his newly published book. William is a keen reader of our newsletter, and wanted to thank me for publishing it. (I am not hinting that I expect gifts from my readers ;-) Being a technical bookworm, I was quite excited to get my hands on the book, although I was not too excited about the title. How interesting could a book about RMI be?!?

The book arrived, and I started reading it here and there, and I quickly noticed that it was far more than the typical Java books that only contain a beautified view of the Java APIs. It is actually a book about distributed computing, and includes chapters on threads, serialization, and such topics. Infact, only half the book is about RMI, the rest is about writing distributed applications.

I did not have the time to finish reading the book before I went to Germany during March and April 2002. Having two little children to take with, we were pressed for space, so the Java RMI book stayed in South Africa. One of my tasks in Germany was to improve the performance of an application server and naturally, one of the hotspots we looked at was the RMI communication. Remembering Bill's book, I promptly went to a local bookshop and bought another copy (that's how useful I found it!).

One of the things we wanted to look at was the number of bytes transferred over the network from and to the application server. This is where the book came in. It showed me how to specify my own socket factory for RMI and how to link in my own counting stream.

This newsletter is therefore based on ideas gleaned from Java RMI. Some parts of the book are not very advanced, and it has parts of Java in there that do not seem to fit into a book on "Java RMI". Perhaps a different title would have been better. However, there are things in the book that have really helped me improved RMI communication when I did not know where to turn. I therefore recommend the book to both beginners in RMI and those who have already used RMI a bit.

The actual code

When I used these tricks in Germany, I hacked the java.net.Socket class and added support for counting bytes. I did that because it was the easiest way, however, you

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

can get into trouble legally if you ship such a hacked class by mistake. For this newsletter, I wanted to go the "right" way by providing an RMISocketFactory.

The first thing we need if we want to count the bytes is to have two decorators for the input and output streams. To me, just knowing *how many* bytes were flowing past was not enough. I also wanted to be able to open the bytes in a hex editor (yeah, baby, yeah) to see what bytes were passed around. Please meet, the DebuggingInputStream (lots of applause):

```
import java.io.*;
import java.net.Socket;

/**
 * This class counts the number of bytes read by it before
 * passing them on to the next Inputstream in the IO chain.
 * It also dumps the bytes read into a file.
 * Should probably specify a factory for making the file
 * names, however, there is enough stuff to show here without
 * such an extra.
 */
public class DebuggingInputStream extends FilterInputStream {
  // Static data and methods
  private static long totalCount = 0;
  private static long dumpNumber =
    System.currentTimeMillis() / 1000 * 1000;

  private synchronized static String makeFileName() {
    return "dump.read." + dumpNumber++ + ".log";
  }
  public synchronized static long getTotalCount() {
    return totalCount;
  }

  // Non-static data and methods
  private final OutputStream copyStream;
  private long count = 0;

  public DebuggingInputStream(Socket socket, InputStream in)
      throws IOException {
    super(in);
    String fileName = makeFileName();
    System.out.println(socket + " -> " + fileName);
    copyStream = new FileOutputStream(fileName);
  }

  public long getCount() {
    return count;
  }

  public int read() throws IOException {
    int result = in.read();
    if (result != -1) {
      synchronized (DebuggingInputStream.class) {
        totalCount++;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      }
      copyStream.write(result);
      count++;
    }
    return result;
  }
  public int read(byte[] b) throws IOException {
    return read(b, 0, b.length);
  }
  public int read(byte[] b, int off, int len)
      throws IOException {
    int length = in.read(b, off, len);
    if (length != -1) {
      synchronized (DebuggingInputStream.class) {
        totalCount += length;
      }
      copyStream.write(b, off, length);
      count += length;
    }
    return length;
  }
  public void close() throws IOException {
    super.close();
    copyStream.close();
  }
}
```

We have a similar class for the OutputStream. Both these classes contain hardcoded values, like for example the filename generation, the fact that it actually passes the data to a file, etc. Obviously, that is not pretty, and could be refactored.

```
import java.io.*;
import java.net.Socket;

public class DebuggingOutputStream extends FilterOutputStream {
  // Static data and methods
  private static long totalCount = 0;
  private static long dumpNumber =
    System.currentTimeMillis() / 1000 * 1000;

  private synchronized static String makeFileName() {
    return "dump.written." + dumpNumber++ + ".log";
  }

  public synchronized static long getTotalCount() {
    return totalCount;
  }

  // Non-static data and methods
  private final OutputStream copyStream;
  private long count = 0;

  public DebuggingOutputStream(Socket socket, OutputStream o)
      throws IOException {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    super(o);
    String fileName = makeFileName();
    System.out.println(socket + " -> " + fileName);
    copyStream = new FileOutputStream(fileName);
  }

  public long getCount() {
    return count;
  }

  public void write(int b) throws IOException {
    synchronized (DebuggingOutputStream.class) {
      totalCount++;
    }
    count++;
    out.write(b);
    copyStream.write(b);
  }
  public void write(byte[] b) throws IOException {
    write(b, 0, b.length);
  }
  public void write(byte[] b, int off, int len)
      throws IOException {
    synchronized (DebuggingOutputStream.class) {
      totalCount += len;
    }
    count += len;
    out.write(b, off, len);
    copyStream.write(b, off, len);
  }
  public void close() throws IOException {
    super.close();
    copyStream.close();
  }
  public void flush() throws IOException {
    super.flush();
    copyStream.flush();
  }
}
```

Next, let us look at our implementation of Socket, called MonitoringSocket. When you look inside java.net.Socket, you can see that all the calls get delegated to a SocketImpl class. The data member inside Socket is called impl and it is package private, meaning that it can be accessed and changed from other classes in the same package. I know what you're thinking - surely that does not happen?! Yes it does - java.net.ServerSocket sometimes sets the impl data member of Socket to **null**. When we then try and print the socket to the screen in the dump() method, we get a NullPointerException. We therefore have to do some hacking to check whether impl is **null** and if it is, we skip over it. We still want to keep a handle to that socket, because impl might be set to another value later.

The rest of MonitoringSocket is fairly straight forward. We have a monitoring thread that once every 5 seconds dumps the active sockets. Yes, it is again hard-coded, but this is debugging code, not production code.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

We then have a non-static initializer block and two constructors. At compile time, the contents of the non-static initializer blocks are copied into the beginning of the constructors (after the call to **super()**). We only show the two constructors needed for the socket factories, the no-args constructor and the one taking a hostname as String and the port.

We obviously also override the getInputStream() and getOutputStream() methods to return DebuggingInputStream and DebugggingOutputStream instances respectively.

```
import java.io.*;
import java.lang.ref.SoftReference;
import java.lang.reflect.Field;
import java.net.*;
import java.util.*;

public class MonitoringSocket extends Socket {
  // keep a list of active sockets, referenced by SoftReference
  private static final List sockets = new LinkedList();

  private static void dump() {
    System.out.println("Socket dump:");
    System.out.println("------------");
    System.out.println("Total bytes"
       + " read=" + DebuggingInputStream.getTotalCount()
       + ", written=" + DebuggingOutputStream.getTotalCount());
    // print all the sockets, and remove them if the Soft
    // Reference has been set to null.
    synchronized (sockets) {
      Iterator it = sockets.iterator();
      while (it.hasNext()) {
        SoftReference ref = (SoftReference)it.next();
        MonitoringSocket socket = (MonitoringSocket)ref.get();
        if (socket == null)
          it.remove();
        else if (!socket.isImplNull())
          System.out.println(socket);
      }
    }
    System.out.println();
  }

  private static Field socket_impl = null;
  static {
    try {
      socket_impl = Socket.class.getDeclaredField("impl");
    } catch (NoSuchFieldException e) {
      throw new RuntimeException();
    }
    socket_impl.setAccessible(true);
  }
  // Sometimes, the Socket.impl data member gets set to null
  // by the ServerSocket.  Yes, it is ugly, but I did not write
  // the java.net.* package ;-)
  private boolean isImplNull() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  try {
    return null == socket_impl.get(this);
  } catch (Exception ex) {
    return true;
  }
}

static {
  new Thread("Socket Monitor") {
    { setDaemon(true); start(); }
    public void run() {
      try {
        while (true) {
          try {
            sleep(5000);
            dump();
          } catch (RuntimeException ex) {
            ex.printStackTrace();
          }
        }
      } catch (InterruptedException e) {} // exit thread
    }
  };
}

private DebuggingInputStream din;
private DebuggingOutputStream dout;

{ // initializer block
  synchronized (sockets) {
    sockets.add(new SoftReference(this));
  }
}
public MonitoringSocket() {}
public MonitoringSocket(String host, int port)
    throws UnknownHostException, IOException {
  super(host, port);
}

private long getBytesRead() {
  return din == null ? 0 : din.getCount();
}

private long getBytesWritten() {
  return dout == null ? 0 : dout.getCount();
}

public synchronized void close() throws IOException {
  synchronized (sockets) {
    Iterator it = sockets.iterator();
    while (it.hasNext()) {
      SoftReference ref = (SoftReference) it.next();
      if (ref.get() == null || ref.get() == this) {
        it.remove();
      }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
   }
   super.close();
   if (din != null) { din.close(); din = null; }
   if (dout != null) { dout.close(); dout = null; }
  }
  public InputStream getInputStream() throws IOException {
   if (din != null) return din;
   return din =
    new DebuggingInputStream(this, super.getInputStream());
  }
  public OutputStream getOutputStream() throws IOException {
   if (dout != null) return dout;
   return dout =
    new DebuggingOutputStream(this, super.getOutputStream());
  }
  public String toString() {
    return super.toString()
     + " read=" + getBytesRead()
     + ", written=" + getBytesWritten();
  }
}
```

The next job is to find all the places in RMI where sockets are created. The most obvious place is in the ServerSocket, so let us change that first:

```
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class MonitoringServerSocket extends ServerSocket {
  public MonitoringServerSocket(int port) throws IOException {
   super(port);
  }
  public Socket accept() throws IOException {
   Socket socket = new MonitoringSocket();
   implAccept(socket);
   return socket;
  }
}
```

Next, we need to tackle the place where RMI creates sockets. RMI provides the ability to specify your own socket factory inside the java.rmi.server.RMISocketFactory class. The default socket factory provided by Sun is the sun.rmi.transport.proxy.RMIMasterSocketFactory class, and contains logic for reusing sockets. It is quite a sophisticated beast, not something that you want to write an ad-hoc implementation for. We could write our own RMISocketFactory to always create a new socket, but then we are not seeing an accurate reflection of what RMI actually does. I found the best approach (besides simply modifying java.net.Socket) is to extend the default socket factory provided by Sun, but there is a catch: Sun's socket factory delegates the actual creation to another instance of RMISocketFactory, i.e. it is just a Decorator for a plain socket factory. The handle to the decorated object is called initialFactory, so what I did was to make that handle point to an instance of RMISocketFactory that created my MonitoringSocket and

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

MonitoringServerSocket classes. There is another catch that I did not address in my code. Sometimes, when you want to speak RMI from behind a firewall, Sun's socket factory creates a socket that can speak over HTTP or CGI interfaces. I do not cover that case, I only cover normal sockets.

```
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.rmi.server.RMISocketFactory;
import sun.rmi.transport.proxy.RMIMasterSocketFactory;

public class MonitoringMasterSocketFactory
    extends RMIMasterSocketFactory {
public MonitoringMasterSocketFactory() {
   initialFactory =  new RMISocketFactory() {
     public Socket createSocket(String host, int port)
        throws IOException {
       return new MonitoringSocket(host, port);
     }
     public ServerSocket createServerSocket(int port)
        throws IOException {
       return new MonitoringServerSocket(port);
     }
   };
  }
}
```

How do you activate this socket factory? At some point at the start of your program, you have to say:

```
RMISocketFactory.setSocketFactory(
  new MonitoringMasterSocketFactory());
```

I'll include some sample code for those who need it:

```
import java.rmi.*;
import java.util.Map;

public interface RMITestI  extends Remote {
  String NAME =  "rmitest";
  Map getValues(Map old) throws RemoteException;
}
```

The implementation of that interface is shown here. It is just a dumb example of sending data backwards and forward, where the size of data being passed grows exponentially:

```
import java.io.IOException;
import java.rmi.*;
import java.rmi.server.*;
import java.util.*;

public class RMITest  extends UnicastRemoteObject
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    implements RMITestI {
  private final Map values = new HashMap();

  public RMITest() throws RemoteException {}

  public Map getValues(Map old) {
    synchronized (values) {
      values.putAll(old);
      return values;
    }
  }

  public static void main(String[] args) throws IOException {
    RMISocketFactory.setSocketFactory(
      new MonitoringMasterSocketFactory());
    System.setSecurityManager(new RMISecurityManager());
    Naming.rebind(RMITestI.NAME, new RMITest());
  }
}
```

And lastly, some client code that connects to the RMI Server and executes the method a number of times. You can see the data that gets passed backwards and forwards by looking at the dump files.

```
import java.io.Serializable;
import java.rmi.*;
import java.util.*;

public class RMITestClient {
  public static void main(String args[]) throws Exception {
    System.setSecurityManager(new RMISecurityManager());
    RMITestI test = (RMITestI)Naming.lookup(RMITestI.NAME);
    Map values = new HashMap();
    values.put(new Serializable() {}, "Today");
    for (int i = 0; i < 13; i++) {
      System.out.print('.');
      System.out.flush();
      values.putAll(test.getValues(values));
    }
  }
}
```

When we run this code, on the server side we can now see the following output:

```
Socket[addr=cohiba/1.0.0.1,port=1099,localport=2135] read=0, written=0 ->
dump.written.1034160774000.log
Socket[addr=cohiba/1.0.0.1,port=1099,localport=2135] read=0, written=7 ->
dump.read.1034160775000.log
Socket[addr=cohiba/1.0.0.1,port=2137,localport=2134] read=0, written=0 ->
dump.read.1034160775001.log
Socket[addr=cohiba/1.0.0.1,port=2137,localport=2134] read=7, written=0 ->
dump.written.1034160774001.log
Socket dump:
------------
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
Total bytes read=507, written=539
Socket[addr=cohiba/1.0.0.1,port=2137,localport=2134] read=471, written=301
Socket[addr=cohiba/1.0.0.1,port=1099,localport=2135] read=36, written=238

Socket[addr=cohiba/1.0.0.1,port=2140,localport=2134] read=0, written=0 ->
dump.read.1034160775002.log
Socket[addr=cohiba/1.0.0.1,port=2140,localport=2134] read=7, written=0 ->
dump.written.1034160774002.log
Socket dump:
-----------
Total bytes read=460579, written=403442
Socket[addr=cohiba/1.0.0.1,port=2137,localport=2134] read=471, written=301
Socket[addr=cohiba/1.0.0.1,port=1099,localport=2135] read=36, written=238
Socket[addr=cohiba/1.0.0.1,port=2140,localport=2134] read=460072,
written=402903

Socket dump:
-----------
Total bytes read=652415, written=1052723
Socket[addr=cohiba/1.0.0.1,port=2137,localport=2134] read=471, written=301
Socket[addr=cohiba/1.0.0.1,port=1099,localport=2135] read=36, written=238
Socket[addr=cohiba/1.0.0.1,port=2140,localport=2134] read=651908,
written=1052184

Socket dump:
-----------
Total bytes read=1702402, written=1091048
Socket[addr=cohiba/1.0.0.1,port=2140,localport=2134] read=1701895,
written=1090509

Socket dump:
-----------
Total bytes read=1702402, written=2301608
Socket[addr=cohiba/1.0.0.1,port=2140,localport=2134] read=1701895,
written=2301069

Socket dump:
-----------
Total bytes read=1702402, written=2752353

Socket dump:
-----------
Total bytes read=1702402, written=2752353
```

We can use this MonitoringSocket wherever RMI is used, for example, in EJB servers like JBoss. If the socket factories are too much P.T. for you (P.T. stands for Physical Training, something that computer nerds try to avoid ;-) you can go and hack java.net.Socket and add the code from the MonitoringSocket in there.

I hope this newsletter will be as useful to you as it was to me. If you want to know more about these topics, please look at William Grosso's Java RMI.

A special thanks to William Grosso for all his help in writing custom sockets.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Until the next newsletter ...

Your Java/Design Patterns/Drinking Straw specialist.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 059] - When arguments get out of hand...

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 59th edition of **The Java(tm) Specialists' Newsletter** sent to 5008 Java Specialists in 89 countries. We have broken through the 5000 mark, I am very grateful to all of you for "spreading the word" about this newsletter.

The original topic of this newsletter was "Verrrrrry looooong Strings and other things". One of my subscribers from Poland suggested that the topic did not really represent the content, so I renamed it.

A few monts ago, my daugter pulled off one of my keys from my Asus notebook. Te irritating part is tat te company from wom I bougt te notebook went bankrupt, and it is now really ard to get spare parts in Sout Africa. I would tell you wic key it is if I could ... ;-)

I started typing this newsletter whilst sitting in a coach, speeding across the English country-side. I spent three days in October at a large company in London, presenting my Design Patterns Course and followed that up with a quick visit to mom-in-law in Sidmouth. Whilst in London, I met up with a number of my newsletter subscribers from the company, and they gave me a memorable introduction to English beverages. Thanks especially to Joe for organising it. That Design Patterns Course was the best I've had so far, gauged by the discussions we had around the various subtleties of the patterns. I am also looking at running a **Design Patterns Course in Munich, Germany** some time at the end of December 2002 or January 2003. We

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

have already identified a very nice hotel where we would like to run the course. Please email me if you would like to attend.

The British Broadcast Corporation (BBC) lists 50 Places to See Before You Die. Cape Town, the city in which I live, is listed as number 5, after The Grand Canyon, Great Barrier Reef, Florida and South Island. I hope that you have been convinced that you should come for a visit :-) And while you pack your suitcase, pop me an email and let me know when you're arriving. My father-in-law is a very reliable taxi-driver who will take you all around Cape Town and will not rip you off. Of course, I also love meeting with my subscribers as time permits. And lastly, if you want to use the trip as a tax break, why not throw in some training with "The Java Specialists"? If you seriously want to do that, please send me an email.

Verrrrrry looooong Strings and other things

One of the things I enjoy doing, is seeing how far I can take things. For example, the rev counter of my Alfa Romeo is limited to 6000 revolutions per minute (revs). The question is, for each of the gears, what is the km/h equivalent of 6000 revs? I found out the hard way on the first day when I had just gotten the car. I was sitting next to a Mercedes at a traffic light, waiting for it to change to green. As it did, I threw in the first gear, charging it up to 60km/h, smugly watching the Mercedes in my rearview mirror, then threw in the second gear, and promptly got the accelerator stuck underneath the floormat! I discovered that the second gear goes up to 100 km/h at 6000 revs. The merc driver must have thought: "Typical show-off Alfa Romeo driver!" After some desperate moments, I managed to pull the floormat back, much to my relief, as I was heading for two speedbumps. Fortunately the breaks are extremely responsive. I have since discovered that gear #3 goes up to 140km/h, and gear #4 is not actually limited to 6000 revs (I don't know why), but it hits 6000 revs at about 180km/h. I do not know at what point gear #5 hits 6000 revs...

Back to Java. At what point do the wheels start falling off? Here are some limitations of the Java Virtual Machine, taken from the VM Spec. They are implicit due to the data structure for the class file format. Consider these restrictions as the rev counter in your JVM:

The per-class or per-interface constant pool is limited to 65535 entries by the 16-bit constant_pool_count field of the ClassFile structure. This acts as an internal limit on the total complexity of a single class or interface.
The amount of code per non-native, non-abstract method is limited to 65536 bytes by the sizes of the indices in the exception_table of the Code attribute, in the LineNumberTable attribute, and in the LocalVariableTable attribute.
The greatest number of local variables in the local variables array of a frame created upon invocation of a method is limited to 65535 by the size of the max_locals item of the Code attribute giving the code of the method. Note that values of type long and double are each considered to reserve two local variables and contribute two units toward the max_locals value, so use of local variables of those types further reduces this limit.
The number of fields that may be declared by a class or interface is limited to 65535 by the size of the fields_count item of the ClassFile structure. Note that the value of the fields_count item of the ClassFile structure does not include fields that are inherited from superclasses or superinterfaces.
The number of methods that may be declared by a class or interface is limited to 65535 by the size of the methods_count item of the ClassFile structure. Note that

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

the value of the methods_count item of the ClassFile structure does not include methods that are inherited from superclasses or superinterfaces.
The number of direct superinterfaces of a class or interface is limited to 65535 by the size of the interfaces_count item of the ClassFile structure.
The size of an operand stack in a frame is limited to 65535 values by the max_stack field of the Code_attribute structure. Note that values of type long and double are each considered to contribute two units toward the max_stack value, so use of values of these types on the operand stack further reduces this limit.
The number of local variables in a frame is limited to 65535 by the max_locals field of the Code_attribute structure and the 16-bit local variable indexing of the Java virtual machine instruction set.
The number of dimensions in an array is limited to 255 by the size of the dimensions opcode of the multianewarray instruction and by the constraints imposed on the multianewarray, anewarray, and newarray instructions.
The number of method parameters is limited to 255 by the definition of a method descriptor, where the limit includes one unit for this in the case of instance or interface method invocations. Note that a method descriptor is defined in terms of a notion of method parameter length in which a parameter of type long or double contributes two units to the length, so parameters of these types further reduce the limit.
The length of field and method names, field and method descriptors, and other constant string values is limited to 65535 characters by the 16-bit unsigned length item of the CONSTANT_Utf8_info structure. Note that the limit is on the number of bytes in the encoding and not on the number of encoded characters. UTF-8 encodes some characters using two or three bytes. Thus, strings incorporating multibyte characters are further constrained.

Ok, that list sounds quite exhausting. Imagine typing in more than 255 parameters! What's the point of worrying about these restrictions?

When I think about going beyond these restrictions, I am thinking about **auto-generated code**, one of the many perls mentioned in The Pragmatic Programmer: From Journeyman to Master. No-one is going to type in a method that contains more than 65535 instructions. At least, if they did, I would be happy for their code to fail. However, you might want to write some code that generates code automatically, and who knows, perhaps for some reason, you want to let a method contain that many instructions.

More than 255 parameters (rule 10 above)

I do not want to cover all the restrictions, just a few. The rest will be left as an exercise to the reader.

The first one I want to look at is the restriction to only allow 255 parameters. This includes the **this** parameter that is automatically passed into non-static methods. Therefore, non-static methods are actually limited to 254 parameters. What happens when you auto-generate a method with 5000 parameters?

```
import java.io.*;

public class ManyParametersGenerator {
  public static void main(String[] args)  throws IOException {
    int LENGTH = Integer.parseInt(args[0]);
    System.out.println("Creating java file with " + LENGTH +
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      " parameters");
    // First, we generate a class with many parameters in a method
    PrintStream out = new PrintStream(
      new BufferedOutputStream(
        new FileOutputStream("ManyParameters.java")));
    out.println("public class ManyParameters {");
    out.println("  public int f(");
    for(int i=0; i<LENGTH; i++) {
      out.print("     int i" + i);
      if (i == LENGTH-1) out.println(") {");
      else out.println(",");
    }
    out.println("    int j = 0 ");
    for(int i=0; i<LENGTH; i++) {
      out.println("     + i" + i);
    }
    out.println("     ;");
    out.println("    System.out.println(j);");
    out.println("    return j;");
    out.println("  }");
    out.println("}");
    out.close();
    // Second, we generate a class that tests our strange class
    out = new PrintStream(
      new BufferedOutputStream(
        new FileOutputStream("ManyParametersTest.java")));
    out.println("public class ManyParametersTest {");
    out.println("  public static void main(String[] args) {");
    out.println("    ManyParameters mp = new ManyParameters();");
    out.println("    int j = mp.f(");
    for(int i=0; i<LENGTH; i++) {
      out.print("     " + (i + 10));
      if (i == LENGTH-1) out.println(");");
      else out.println(",");
    }
    // we also calculate what "j" should actually be
    int j = 0;
    for(int i=0; i<LENGTH; i++) {
      j += i+10;
    }
    out.println("    System.out.println(j);");
    out.println("    System.out.println(" + j + ");");
    out.println("  }");
    out.println("}");
    out.close();
  }
}
```

All code in this newsletter was compiled and run with Sun Microsystems' JDK 1.4.1_01 on Windows 2000 Professional, unless stated otherwise.

Let us run this with 10, 100, 254, 255 and 5000 parameters, followed by a compile and execution of our test classes:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Run with 10 parameters:

> java ManyParametersGenerator 10
Creating java file with 10 parameters
> javac ManyParameters.java ManyParametersTest.java
> java ManyParametersTest
145
145
145

Run with 100 parameters:

> java ManyParametersGenerator 100
Creating java file with 100 parameters
> javac ManyParameters.java ManyParametersTest.java
> java ManyParametersTest
5950
5950
5950

Run with 254 parameters:

> java ManyParametersGenerator 254
Creating java file with 254 parameters
> javac ManyParameters.java ManyParametersTest.java
> java ManyParametersTest
34671
34671
34671

Run with 255 parameters:

```
> java ManyParametersGenerator 255
Creating java file with 255 parameters
> javac ManyParameters.java ManyParametersTest.java
> java ManyParametersTest
Exception in thread "main" java.lang.VerifyError: (class: Many
ParametersTest, method: main signature: ([Ljava/lang/String;)V
) Signature (IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII` L#
```

Let's take a quick sanity check break here. It compiles fine, but the class verifier blows up, because there are 256 parameters (if we include the **this** pointer that implicitely gets sent to the method). Surely, the number of parameters could also be checked by the compiler? So, why do it in the verifier? The reason is that if it were **only** done in the compiler, I could write my own compiler that generated methods with more than 255 parameters, and that could cause your JVM to break. Think applets. However, it would make sense to me to also restrict the standard compiler to only allow 255 parameters. Perhaps *they* thought that no-one would be crazy enough to use more than 255 parameters anyway...

Run with 5000 parameters:

```
> java ManyParametersGenerator 5000
Creating java file with 5000 parameters
> javac ManyParameters.java ManyParametersTest.java
> java ManyParametersTest
The system is out of resources.
Consult the following stack trace for details.
java.lang.StackOverflowError
  at com.sun.tools.javac.v8.comp.Attr.attribExpr(Attr.java:279)
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  at com.sun.tools.javac.v8.comp.Attr.visitBinary(Attr.java:965)
  at com.sun.tools.javac.v8.tree.Tree$Binary.accept(Tree.java:1014)
  at com.sun.tools.javac.v8.comp.Attr.attribTree(Attr.java:256)
  at com.sun.tools.javac.v8.comp.Attr.attribExpr(Attr.java:279)
etc.
```

What do we learn from this experience? Not very much. We hear the Java engine revving a bit, but besides that, all we notice is that the engine is limited to 6000 revs. However, we do now know how far we can push Java. Not very.

Long constant Strings (rule 11 above)

Constant Strings must be shorter than 65536 characters. This also means that variable names, method names and field names must be less than 65536 characters. Oh no! Again, usually the only conceivable time when you could end up with a String that long is with automatically generated code, although I know of one company that embedded their SQL queries in constant Strings and then hit this limit!

```
import java.io.*;

public class BigStringGenerator {
  public static void main(String[] args) throws IOException {
    int LENGTH = Integer.parseInt(args[0]);
    System.out.println("Creating java file with string of length " + LENGTH);
    PrintStream out = new PrintStream(
      new BufferedOutputStream(
        new FileOutputStream("BigString.java")));
    out.println("public class BigString {");
    out.print("  public String big = ");
    out.print("\"");
    for(int i=0; i<LENGTH; i++) {
      out.print((char)((i%26)+'A'));
    }
    out.print("\"");
    out.println(";");
    out.println("}");
    out.close();
    out = new PrintStream(
      new BufferedOutputStream(
        new FileOutputStream("BigStringTest.java")));
    out.println("public class BigStringTest {");
    out.println("  public static void main(String[] args) {");
    out.println("    try {");
    out.println("      BigString bs = new BigString();");
    out.println("      System.out.println(bs.big.length());");
    out.println("    } catch(Throwable t) { System.err.println(t); }");
    out.println("  }");
    out.println("}");
    out.close();
  }
}
```

Let us run this with Strings of size 65535, 65536, 600000 and 6000000, followed by a compile and execution of our test classes:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Run with String of length 65535:

```
> java BigStringGenerator 65535
Creating java file with string of length 65535
> javac BigString.java BigStringTest.java
> java BigStringTest
65535
```

Run with String of length 65536:

```
> java BigStringGenerator 65536
Creating java file with string of length 65536
> javac BigString.java BigStringTest.java
> java BigStringTest
Exception in thread "main" java.lang.ClassFormatError: BigString (Illegal constant
pool type)
    at java.lang.ClassLoader.defineClass0(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:502)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:123)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:250)
    at java.net.URLClassLoader.access$100(URLClassLoader.java:54)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:193)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:186)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:299)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:265)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:255)
    at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:315)
```

Run with String of length 600000:

```
> java BigStringGenerator 600000
Creating java file with string of length 600000
> javac BigString.java BigStringTest.java
> java BigStringTest
Exception in thread "main" java.lang.ClassFormatError: BigString (Illegal constant
pool type)
    at java.lang.ClassLoader.defineClass0(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:502)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:123)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:250)
    at java.net.URLClassLoader.access$100(URLClassLoader.java:54)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:193)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:186)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:299)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:265)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:255)
    at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:315)
```

Run with String of length 6000000:

```
> java BigStringGenerator 6000000
Creating java file with string of length 6000000
> javac BigString.java BigStringTest.java
The system is out of resources.
Consult the following stack trace for details.
java.lang.OutOfMemoryError
> java BigStringTest
Exception in thread "main" java.lang.ClassFormatError: BigString (Truncated class
file)
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Ok, that was pushing it a bit. However, I am a bit worried that you cannot change the maximum heap memory when you run the javac tool.

What are the missing Fields? (rule 4 above)

I'm confused (a bit). In rule 4, we see that the number of fields is limited to 65535. However, when I pushed the gas pedal, I found that I was *only* allowed 65521 data members (in the JDK 1.4.1_01). Are there hidden fields, or did the compiler writers not check the boundary conditions properly? The JDK 1.3.1_03 contains a bug that only allows you to have 12996 data members, so it is a 404% improvement. Have a look at the following code:

```
import java.io.*;

public class BigClassGenerator {
  public static void main(String[] args)  throws IOException {
    int LENGTH = Integer.parseInt(args[0]);
    System.out.println("Creating java file with " + LENGTH +
      " data members");
    PrintStream out = new PrintStream(
      new BufferedOutputStream(
        new FileOutputStream("BigClass.java")));
    out.println("public class BigClass {");
    for(int i=0; i<LENGTH; i++) {
      out.println("private int a" + i + ";");
    }
    out.println("}");
    out.close();
  }
}
```

Try this on JDK 1.4.1_01 and JDK 1.3.1_03 and see if you get the same results as me. Assuming that JDK 1.4.1_01 is the most correct version of Java we have available at the moment, what are the missing 14 fields? Are there hidden fields? Perhaps hidden static fields?

Classes in a dependency chain

I don't know where this last idea would be knocking against the rules. In case you have made up your mind that the latest and greatest JDK 1.4.1_01 is the answer for all your compilation problems, have a look at this example:

```
import java.io.*;

public class ClassGenerator {
  public static void main(String[] args)  throws IOException {
    int NUMBER_OF_CLASSES = Integer.parseInt(args[0]);
    for (int i=0; i<NUMBER_OF_CLASSES; i++) {
      String filename = "A" + i + ".java";
      System.out.println(filename);
      PrintStream out = new PrintStream(
        new FileOutputStream(filename));
      out.println("public class A" + i + " { private A" +
        ((i+1)%NUMBER_OF_CLASSES) + " other; }");
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    out.close();
   }
  }
}
```

In JDK 1.4.1_01 if you run this with a value of 176 or higher, you get the following result:

```
The system is out of resources.
Consult the following stack trace for details.
java.lang.StackOverflowError
    at java.util.Hashtable.get(Hashtable.java:329)
    at java.util.Properties.getProperty(Properties.java:480)
    at java.lang.System.getProperty(System.java:574)
    at sun.security.action.GetPropertyAction.run(GetPropertyAction.java:66)
    at java.security.AccessController.doPrivileged(Native Method)
    at sun.io.Converters.getDefaultEncodingName(Converters.java:66)
    at sun.nio.cs.StreamDecoder.forInputStreamReader(StreamDecoder.java:69)
    at java.io.InputStreamReader.<init>(InputStreamReader.java:57)
    at com.sun.tools.javac.v8.parser.Scanner.<init>(Scanner.java:139)
    at com.sun.tools.javac.v8.JavaCompiler.parse(JavaCompiler.java:231)
    at com.sun.tools.javac.v8.JavaCompiler.complete(JavaCompiler.java:305)
    ... etc.
```

In JDK 1.3.1_03, I ran the test for 8000 classes, it compiled fine, but took a while to complete. So, don't be lulled into a false sense of security with the new compilers. Your long SQL Queries might now compile, but don't write programs that are too complex ;-)

Until our next newsletter...

Heinz

P.S. Please let me know if you found this newsletter particularly interesting. I have been thinking about this topic since January 2000 when I was flying back from Germany after a serious consulting expedition with my colleague Paul van Spronsen and we started bashing the JVM around while sitting 30'000 feet above the ground. You have just read thoughts that span 3 years :-)

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

..... [Issue 059b] - Follow-up to Loooong Strings

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Thanks to those of you who wrote back after my last newsletter. There are some issues I want to follow-up on.

To start off, the date on the newsletter was incorrect.

Then, the original topic of this newsletter was "Verrrrrry looooong Strings and other things". One of my subscribers from Poland suggested that the topic did not really represent the content as there is no real limit on how long ordinary Strings can be, except for the limit on char array length and physical memory. I renamed it to: "When arguments get out of hand..."

Talking about long constant Strings, many thanks to Frank Peters from Process Management Consulting in Cologne for pointing out the problem with long constant Strings to me at the beginning of June 2002. Apologies for forgetting to thank you in the original newsletter, our email conversation had slipped my conscious mind. Something that Frank pointed out to me in the original discussion was that if you have a unicode character, it might be converted to more than two bytes in certain circumstances, so the limit is actually less than 65535 in some cases.

Some readers also wrote to tell me that the problem with looong constant Strings used to appear with JSP pages generated by Dreamweaver.

Last, but definitely not least, thanks to Bjorn Carlin for sending me the secret to the 14 missing data members. Thank you also to John Bester from South Africa and Juan C. Valverde from ArtInSoft in Costa Rica for sending me their ideas.

The missing 14 fields

The limit on the fields is actually the limit in the constant pool where the names of the fields are stored. Bjorn told me that the 9th and 10th bytes of the class file tell you how many constant fields are in the class. He also sent me output from a little program he wrote that dumped the constant pool from a class.

Here is a piece of code that tells you the size of the constant pool for a classfile:

```
import java.io.*;

public class ConstantPoolSize {
  static void skip(InputStream in, int skip) throws IOException {
    while(skip > 0) {
      skip -= in.skip(skip);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
  }

  public static void main(String[] args) throws IOException {
    DataInputStream din = new DataInputStream(
      new FileInputStream(args[0]));
    skip(din, 8);
    int count = ((din.readByte() & 0xff) << 8)
      | (din.readByte() & 0xff);
    System.out.println(count);
    din.close();
  }
}
```

I'll let you figure out the reason for skip() yourself ;-)

Run it against a small class such as:

```
public class Small {}
```

You will get different results depending with which debugging info options you compiled Small.java. The default compile will tell you that there are 13 entries in Small.class. If you add one field, you will have 15 entries, one for the name of the field, and one for the type of the field. For example, if you add an int a0;, you will have entries "I" and "a0".

You can compile BigClass with the -g:none option, then you can squeeze more fields in. In addition, if you call one field "I" and another "Code", you can get two more fields into your class. "I" is the type of int defined in the constants table. I don't know where "Code" comes from, but it works. The maximum number of fields I've therefore managed to squeeze in like this was 65526.

Now that's really overrevving Java ;-)

Heinz

P.S. Probably the worst mistake in the newsletter was that I said my rev counter was limited to 6000 revolutions per minute. That of course is nonsense. It is limited to 7000 revs. I checked today. When I wrote the newsletter, I did not feel like going to the garage to look. Besides, at 1:00am that might have made my neighbours unhappy ;-)

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 060] - Nulling variables and garbage collection

Author: Jack Shirazi

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 60th edition of **The Java(tm) Specialists' Newsletter** sent to 5150 Java Specialists in 91 countries. Thank you so much for your support this year, writing this newsletter, and especially reading your replies, has been an absolute pleasure. One thing that excites me is seeing famous names pop up on my reader list. One of these, Jack Shirazi, a household name in Java performance circles and author of Java Performance Tuning, has been watching our newsletter for about a year and Jack and I have often had interesting discussions about the finer details of Java performance. This month Jack decided to interview me, you can gladly read the interview on his website if you would like to.

Today Jack sent me an article for our newsletter, and I am honoured to publish it. Thanks Jack for this excellent article, we all really appreciate the time and effort you took in writing it for us.

After mentioning in my last newsletter how fantastic South Africa is, I have seen a dramatic strengthening of our currency. Ours is the best performing currency against the US$ in the last 12 months. I have come to the conclusion that someone on my mailing list has so much money that they can influence the economy of our country ;-)

Enough of my musing, over to Jack Shirazi:

Jack Shirazi: Nulling variables and garbage collection

At JavaPerformanceTuning.com, we have a monthly newsletter which includes a roundup of recent performance related discussions from various discussion groups. (We also have many more columns, covering almost all recent Java performance related activity, including interviews - last month we interviewed none other than our excellent host, Heinz). The discussions we report are quite interesting, and sometimes a really fascinating one pops up.

One such discussion at Javagaming.org considered whether setting variables to null helps garbage collection in any significant way. After a little inconclusive discussion, a Sun engineer jumped in with a really interesting example (Javagaming.org is a Sun owned discussion board, run by a couple of Sun engineers, and other Sun engineers sometimes lurk around).

This engineer pointed out that if Eden was full, and an object was about to be created, then if sufficient space could be freed in Eden, the object could be created

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

there with no more work. But if sufficient space could NOT be freed, then existing objects in Eden had to be promoted to old space before the new object could be created. It could be that the application was finished with an object, but it had not yet gone out of scope, so could not be immediately garbage collected. Explicitly nulling the variable referencing that object would make a difference.

For those of you not up to spec on HotSpot GC, the heap is divided into several spaces: Eden and two survivor spaces (the three spaces collectively make up the young generation space) and an old generation space. Objects are created in Eden, and most objects die there and are reclaimed quickly. The two survivor spaces are for copying live objects so that young objects can remain in the young generation space for a time. If objects get too old, or young generation space gets full up, objects get promoted to the old generation space. If you need more detail, try these two URLs: Java World and Sun Microsystems.

This being a gaming discussion board, the code example the engineer gave was relevant to animation, and a variation of his example follows here.

```
import java.awt.Image;
import java.awt.image.BufferedImage;

public class ImageGarbage1 {
  private static Image img;    //IMPORTANT 1
  public static void main(String args[]) {
    // Get one numerical argument which specifies size of image
    int imageSize = Integer.parseInt(args[0]);
    long startTime = System.currentTimeMillis();
    int imgIndex = 0;
    long loopIndex = 0;
    while (true) {
      //img = null;   //IMPORTANT 2
      // Create an image object
      img = new BufferedImage(imageSize, imageSize,
        BufferedImage.TYPE_INT_RGB);
      long endTime = System.currentTimeMillis();
      ++loopIndex;
      // We print stats for every two seconds
      if (endTime - startTime >= 2000) {
        // Images created and disposed if per second
        long ips = (loopIndex / ((endTime - startTime) / 1000));
        System.out.println("IPS = " + ips);
        startTime = endTime;
        loopIndex = 0;
      }
    }
  }
}
```

Basically, this class repeatedly creates and discards images, measuring the rate of object creation, and printing that rate every couple of seconds. A single command line argument allows the size of the object to be specified.

On my desktop, I get the following results on running this:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
COMMAND: java -version
java version "1.4.1_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.1_01-b01)
Java HotSpot(TM) Client VM (build 1.4.1_01-b01, mixed mode)

COMMAND: java ImageGarbage1 361
IPS = 14
IPS = 18
IPS = 19
IPS = 19
```

Your machines are probably a darn sight more powerful than my old desktop [hk: goodness me, even my 2 year old notebook is faster ;-], so I'm sure you'll get larger numbers. You need to control-C the process to stop it, otherwise it runs forever. Now let's look at what happens if I uncomment the line labelled "IMPORTANT 2", which simply adds a null assignment to the static variable before creating the image object:

```
COMMAND: java ImageGarbage2 361
IPS = 116
IPS = 160
IPS = 164
IPS = 165
IPS = 151
```

Yes, that's right, the null assignment made the test run an order of magnitude faster! Well I guess that's a pretty definitive answer to the question of whether nulling variables makes a difference. What is happening in the first test (ImageGarbage1) is that the image is created in Eden, and when the next image is created, the first image is still hard referenced by the application, so the GC has to promote the object to the old generation to make space for the new object. In the second test (ImageGarbage2), the inserted null reference means that the existing image in Eden is not referenced by the application, and can be discarded.

However, before you go off changing your coding style, I should point out that this is a specially constructed case. The image size was carefully chosen to fill Eden. Look what happens when we increase the image size slightly, even with the null assignment:

```
COMMAND: java ImageGarbage2 362
IPS = 13
IPS = 18
IPS = 18
IPS = 19
```

In this last test, the image is too big to fit into the young generation, and gets created directly in the old generation each time. Which means that a full mark-sweep GC of the old generation is needed to reclaim the space, rather than the much faster copying GC of the young generation. And if we choose a much smaller size, so that several images fit into Eden, then each subsequent image assignment releases the reference to the previous image, so allowing them all to be collected in Eden:

```
COMMAND: java ImageGarbage1 100
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

IPS = 759
IPS = 1487
IPS = 1490
IPS = 1490

COMMAND: java ImageGarbage2 100
IPS = 810
IPS = 1611
IPS = 1621
IPS = 1613

Let's make one more change. I'll go back to the original version of the test, with the null assignment commented out. But this time I'll change the static variable definition (the line labelled "IMPORTANT 1") into a local variable in the main() method. Now look what happens when I run the test with the image just fitting into Eden:

COMMAND: java ImageGarbage3 361
IPS = 113
IPS = 157
IPS = 165
IPS = 166
IPS = 156

We get back to the same performance as the test with the null assignment, even though we are NOT making that null assignment this time. In this case, the compiler is intelligent enough to work out the scope of the image object, and has dereferenced it before the next assignment, so allowing it to be reclaimed before the next image is created.

Finally, putting together what we've seen, it's easy to work out that Eden's size was the real problem with the original test. So let's get back to that original case, and simply alter the size. Here I simply set the initial heap size larger than the default, which makes Eden proportionately larger:

COMMAND: java -Xms16M ImageGarbage1 361
IPS = 78
IPS = 129
IPS = 120
IPS = 131

And the same test with the slightly larger image, that previously had to be created in old generation space:

COMMAND: java -Xms16M ImageGarbage1 362
IPS = 73
IPS = 120
IPS = 121
IPS = 121

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

So there is no real need to change your coding style. Tuning the garbage collection is probably a more sensible solution. Does nulling variables improve garbage collection? Maybe, sometimes.

Jack Shirazi is the author of "Java Performance Tuning" (O'Reilly), and director of JavaPerformanceTuning.com. JavaPerformanceTuning.com lists three thousand performance tuning tips and publishes a monthly newsletter with all the latest Java performance news, tips, discussions, and more. See http://www.JavaPerformanceTuning.com/

--Jack Shirazi
jack@JavaPerformanceTuning.com
http://www.JavaPerformanceTuning.com

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 061] - Double-checked locking

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 61st edition of **The Java(tm) Specialists' Newsletter** sent to 5261 Java Specialists in 91 countries. Since my last newsletter I had the fortune of celebrating yet another birthday - try guess how old I am ;-)

Has your application ever caused an OutOfMemoryError? Would you like to know how you can receive a warning before that actually happens? Send me an email

2003 - Europe Design Patterns Tour Early in 2003 I will be travelling to Europe to present my Design Patterns Course in various countries. This 3 day course will teach

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

you to be more effective as a Java/C++ programmer by examining many of the Gang-Of-Four Design Patterns that will help you write code that is reusable. At the very least, it will teach you a richer vocabulary for bringing your design ideas across to your colleagues. No more banging your head against the wall because you just cannot think of a good name for what you are trying to describe. The dates for the tour have not been fixed yet, but look to happen towards the end of January or beginning of February 2003.

Estonia looks like a good possibility with one bank already interested in an in-house course.

Munich is one of my favourite places in Germany, with lots of jovial Bavarians, and also some Germans. I would like to offer an open course here for anyone to attend. Please send me an email if you are thinking of attending.

London I will pass through on my way to the other cities. Do you work in London and are you eager to learn the new skills that Design Patterns will give you? I am specifically looking for companies that want to teach this skill to a number of their employees. I take a maximum of 12, a minimum of 8 for an in-house course. Please send me an email if you would like to host such an in-house course.

Double-checked Locking *yawn*

No newsletter writer can claim that he is successful if he has not at least once written about double-checked locking and how it is theoretically broken. It is the ideal topic to write about. When you read my theoretical proofs, you will think I am very clever, thus enabling me to raise my rates next time you ask for help. At the same time, the problems described only happen on a fast multi-processor machine, and it is unlikely that you have one of *those* standing at home, so you cannot even contradict me.

Let's take the Singleton pattern, described in the sample chapters of my Design Patterns Course (this is quite a large file due to the cool illustrative comics). If a Singleton does not need to be extendible, I write it like this:

```
public class Singleton {
  private final static Singleton instance = new Singleton();
  public static Singleton getInstance() { return instance; }
  private Singleton() {}
  // other methods that you would need to do the actual work
}
```

The class will be loaded when it is needed at which point the instance field is created.

Another way of writing the Singleton is:

```
public class Singleton {
  private static Singleton instance;
  public synchronized static Singleton getInstance() {
    if (instance == null) {
      instance = new Singleton();
    }
    return instance;
  }
  private Singleton() {}
  // other methods that you would need to do the actual work
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

The advantage of that approach is that the Singleton instance is only created when you first use it. However, the class loader would usually take care of that aspect anyway, so we do not really win anything by complicating our lives like this. It does not win us anything, but at least it works.

The problem with the new Singleton is that every time you call getInstance(), you will have to synchronize, even though you only really should need to the first time. Clever people have thus come up with this pattern:

```
public class Singleton {
  private static Singleton instance;
  public static Singleton getInstance() {
    if (instance == null) {
      synchronized(Singleton.class) {
        if (instance == null) {
          instance = new Singleton();
        }
      }
    }
    return instance;
  }
  private Singleton() {}
  // other methods that you would need to do the actual work
}
```

That mechanism is commonly called "Double-Checked Locking" or DCL for short. It is broken, at least in theory. In theory what may happen is that due to byte reordering, before the first thread has finished setting up the Singleton, i.e. has finished calling the constructor of the Singleton, the second thread already has a handle to it. In theory, therefore, it is possible that a thread can get a handle to a half-baked object using this mechanism.

I have tried to gather some evidence that with the current JVM implementations the problem can occur. I have written test code and executed it on a friend's multi-processor. I was not successful in proving my assertions. I have spoken to a number of authors who have written about this problem, and they also do not have firm evidence that this occurs in the real world. I therefore offer no proof or even claim that what I wrote about in this newsletter is true.

That is really all I have to say about the matter. Avoid the double-checked locking to avoid synchronization. In JDK 1.5 they are tidying up the memory model thus showing more clearly that DCL is broken. In the meantime, try not to be too clever ;-)

Here is a challenge: I have not found a Singleton that is extendable in such a way that at compile time it is checked that there is only ever one instance. During my Design Patterns Course, I usually spend quite a bit of time on the Singleton, discussing things like why it is a pattern to avoid, what the differences between a Singleton and class methods are, etc. One of the issues we spend some time thinking about is the whole issue of making the Singleton extendable - I call it a polymorphic Singleton.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

That's it for this week.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 062] - The link to the outer class

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 62nd edition of **The Java(tm) Specialists' Newsletter** sent to 5450 Java Specialists in 91 countries.

Here I am, sitting in my shorts on a beautiful starry night on our balcony, listening to the Guinea Fowls in our trees, trying to churn out yet another newsletter to appeal to your cerebral impulses. Guinea Fowls make a very strange sound, which is extremely annoying to some, and music to others. Fortunately for me (and the Guinea Fowls), I like their squawking. From the sunburnt-southern-hemispherer to all eggnog-drinking-northern-hemispherers: Relax, the days are getting longer again :-)

Don't you just love those Out-Of-Office messages? Boy, am I going to get a lot of those with this newsletter...

2003 - Europe Design Patterns Tour - please look at my previous newsletter.

The link to the outer class

Consider the following classes:

```
public abstract class Insect {
  public Insect() {
    System.out.println("Inside Insect() Constructor");
    printDetails();
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public void printDetails() {
    System.out.println("Just an insect");
  }
}

public class Beetle extends Insect {
  private final int legs;
  public Beetle(int legs) {
    System.out.println("Inside Beetle() Constructor");
    this.legs = legs;
  }
  public void printDetails() {
    System.out.println("The beetle has " + legs + " legs");
    if (legs < 6) {
      System.out.println("Ouch");
    }
  }
}

public class BeetleTest {
  public static void main(String[] args) {
    Beetle sad_bug = new Beetle(5); // lost one leg in an
                                    // argument with his wife
    Beetle happy_bug = new Beetle(6); // the wife bug ;-)
  }
}
```

Stop for a moment and think of what the effect would be of running BeetleTest. Don't read further until you have decided what would happen.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

I hope you didn't peep :-) Here is the output:

```
Inside Insect() Constructor
The beetle has 0 legs
Ouch
Inside Beetle() Constructor
Inside Insect() Constructor
The beetle has 0 legs
Ouch
Inside Beetle() Constructor
```

Yes, even though legs was **final**, we were able to access it before it was initialised. What is more, we are able to call the subclass' methods from the constructor of the superclass, before the subclass had been initialised! This should come as no surprise to you, since by being subscribed to **The Java(tm) Specialists' Newsletter** you would be classed as a Java Specialist. Yes? No ... ?

But, the plot thickens. Look at the following class:

```
public class NestedBug {
  private Integer wings = new Integer(2);
  public NestedBug() {
    new ComplexBug();
  }
  private class ComplexBug extends Insect {
    public void printDetails() {
      System.out.println(wings);
    }
  }
  public static void main(String[] arguments) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    new NestedBug();
  }
}
```

When we run this code, we get a NullPointerException:

```
Inside Insect() Constructor
java.lang.NullPointerException
  at NestedBug.access$0(NestedBug.java:2)
  at NestedBug$ComplexBug.printDetails(NestedBug.java:8)
  at Insect.<init>(Insect.java:4)
  at NestedBug$ComplexBug.<init>(NestedBug.java:6)
  at NestedBug.<init>(NestedBug.java:4)
  at NestedBug.main(NestedBug.java:12)
Exception in thread "main"
```

A friend of mine once had this problem, so my first thought was that somehow, because wings was null, we ended up with a NullPointerException when printing it. That explanation did not make sense, because calling toString() on a null pointer is supposed to just return null. My friend changed his code to the following:

```
public class NestedBug2 {
  private Integer wings = new Integer(2);
  public NestedBug2() {
    new ComplexBug();
  }
  private class ComplexBug extends Insect {
    public void printDetails() {
      if (wings != null) { // line 8
        System.out.println(wings);
      }
    }
  }
  public static void main(String[] arguments) {
    new NestedBug2();
  }
}
```

Sadly, this does not make the NullPointerException go away:

```
Inside Insect() Constructor
java.lang.NullPointerException
  at NestedBug2.access$0(NestedBug2.java:2)
  at NestedBug2$ComplexBug.printDetails(NestedBug2.java:8)
  at Insect.<init>(Insect.java:4)
  at NestedBug2$ComplexBug.<init>(NestedBug2.java:6)
  at NestedBug2.<init>(NestedBug2.java:4)
  at NestedBug2.main(NestedBug2.java:14)
Exception in thread "main"
```

But wait! The line with the NullPointerException is the line that simply checks whether wings is null!? Can the mere act of checking whether wings is **null** itself cause a NullPointerException? In this case it appears that it can! Or is the actual mistake on line 2? What is this access$0 method?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

To understand this strange behaviour, we have to understand what the constructor of the inner class consists of. The easiest way of doing this is to use JAD and decompile the class file with the -noinner option:

```
jad -noinner NestedBug2$ComplexBug.class

class NestedBug2$ComplexBug extends Insect {
  NestedBug2$ComplexBug(NestedBug2 nestedbug2) {
    this$0 = nestedbug2;
  }
  public void printDetails() {
    if (NestedBug2.access$0(this$0) != null)
      System.out.println(NestedBug2.access$0(this$0));
  }
  private final NestedBug2 this$0; /* synthetic field */
}
```

We also need to look at NestedBug2.class:

```
jad -noinner NestedBug2.class

public class NestedBug2 {
  public NestedBug2() {
    wings = new Integer(2);
    new NestedBug2$ComplexBug(this);
  }
  public static void main(String arguments[]) {
    new NestedBug2();
  }
  static Integer access$0(NestedBug2 nestedbug2) {
    return nestedbug2.wings;
  }
  private Integer wings;
}
```

Again, I must urge you to spend a few minutes thinking about this. Step through what would happen in your head or scribble on a piece of paper, but try to understand. When printDetails() is called by Insect>, the handle to outer class (this$0) has not yet been set in the ComplexBug class, so that class passes **null** to the access$0 method, which of course causes a NullPointerException when it tries to access nestedbug2.wings as we would expect.

I bet that many developers have run into this problem, but being under typical time pressure would have just coded around it until they got it working, without taking the time to think about what is happening.

This experience leads us to some questions:

- Is `this$0` a new type of keyword you did not know about?
- What happens when your inner class already contains a variable this$0?
- Is calling a method from a constructor a good idea in the first place?
- When, if ever, are inner classes a good idea?

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

I will leave you with these questions to answer yourself over the festive season. For those of you who are actually working, I hope this newsletter has cheered up your workday and that you will be motivated to seek new nuggets of information inside the fascinating Java language.

All the best for the new year. May 2003 bring you new opportunities to learn and grow, not just in the material world :-)

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 062b] - Follow-up and Happy New Year!

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

First of all, I wish you all the best for the year 2003!

Two of our readers, Dimitris Andreou from Greece and another by the name of Snow, pointed out that in JDK 1.4 the NullPointerException has been fixed. Have a look at Sun's website. Try compile the following class with the -target 1.4 option:

```
public class NestedBug3 {
  private Integer wings = new Integer(2);
  public NestedBug3() {
    new ComplexBug();
  }
  private class ComplexBug extends Insect {
    ComplexBug() {
      System.out.println("Inside ComplexBug Constructor");
    }
    public void printDetails() {
      System.out.println(wings);
    }
  }
  public static void main(String[] arguments) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    new NestedBug3();
  }
}
```

When you compile this with -target 1.4 you will only be able to run it with a JVM 1.4 and later. Here is the output:

```
Inside Insect() Constructor
2
Inside ComplexBug Constructor
```

What happens in the inner class? Let's have a quick look:

```
jad -noinner NestedBug3$ComplexBug.class

class NestedBug3$ComplexBug extends Insect {
  NestedBug3$ComplexBug(NestedBug3 nestedbug3) {
    this$0 = nestedbug3;
    super();
    System.out.println("Inside ComplexBug Constructor");
  }
  public void printDetails() {
    System.out.println(NestedBug3.access$000(this$0));
  }
  private final NestedBug3 this$0; /* synthetic field */
}
```

You can seen that the assignment of this$0 and the call to **super**() have been swapped around.

In case you were wondering where to get hold of JAD, I wrote to the author and he told me to tell you to look at http://kpdus.tripod.com/jad.html. I want to hereby publicly thank Pavel Kuznetsov for making this great tool freely available. I have learnt more from this tool than from any other Java tool.

In case you were wondering why I am spending my New Year's Party writing a newsletter, I just want to get this done quickly so that I can put the follow-up into the 2002 folder ;-)

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 063] - Revisiting Stack Trace Decoding

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 63rd edition of **The Java(tm) Specialists' Newsletter** sent to 5726 Java Specialists in 93 countries.

Someone has pressed the fast-play button in my life. It seems like it was last week that I published newsletter #62, yet it is almost a whole month since you last heard from me. At least I know that you appreciate me, by the "How did I get unsubscribed???" messages that appear in my Inbox ;-) This was supposed to be a weekly newsletter, but we are the victims of our own success.

I started this newsletter whilst sitting in a Fish & Chips shop in London, on my way to Estonia. In case your geography is as bad as mine, Estonia is one of the states that was occupied by Russia for a few decades, and became independent in 1991. They revolted against Russia by singing! In my next newsletter I will write a travel report about my experiences in Estonia. I am also thrilled to be able to meet my readers in Estonia, so please send me an email if you are in Tallinn.

When I mentioned Estonia to my friends in South Africa, I got lots of responses of "where's that?". Once I had explained where Estonia was, they usually followed with: "What on earth are you going to be doing *there*?" The answer is that I will be teaching Design Patterns at an Estonian Bank. An in-house course is the best way to learn Design Patterns. I plan to come to Europe again in March, so if you would like to host a course at your company, please send me an email. Just another example that Maximum Solutions goes to the *ends of the earth* for their customers ;-)

Revisiting Stack Trace Decoding

In our fourth newsletter, I mentioned that you could decode the stack trace to find out where the method call that caused your problem came from. At that time already, some readers mentioned that JDK 1.4.x would make this problem easier to solve.

Just to remind you - in the previous newsletter we pushed the stack trace to a StringWriter and then stepped through the String counting the lines until we arrived at the desired line. In JDK 1.4.x Sun Microsystems changed the Throwable class a little bit, making it more useful. One of the things Sun did was keep the elements of the call stack in an array of StackTraceElement objects. This way we can more easily determine what line some method is being called from. One of my readers sent me an example of how this information could be used to enforce security. Perhaps one day I will publish that idea.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

In some of my newsletters I have berated the exceptions. Just to clear the air - I do not have anything against exceptions. I also do not have a problem with *sensible* comments. I just do not like it when Exceptions are ignored. I believe that every Java program should have a central exception mechanism that can receive exceptions that could not be properly handled and that the administrator should know immediately when something went wrong.

I present a new interface called CallDetective that can find out who called us. An interesting twist is that the interface CallDetective contains a static class called Factory that constructs the correct type of CallDetective, depending on the Java Version. For any JVM prior to JDK 1.4, we use the old StackTrace approach discussed in newsletter #4. The idea of having a class inside an interface is not new, we already saw it in newsletter #6.

Here is the CallDetective Interface:

```
/**
 * This interface is used to determine who called us.
 * The implementation does not have to be thread-safe.
 */
public interface CallDetective {
  /**
   * Returns a String representation of who called us,
   * going back depth levels.
   * @param depth must be greater than 0 and may not
   * exceed the call stack depth.
   */
  public String findCaller(int depth);
  public class Factory {
    public static CallDetective makeCallDetective() {
      if ("1.4".compareTo(System.getProperty("java.version")) > 0) {
        return new CallDetective1_3();
      } else {
        return new CallDetective1_4();
      }
    }
  }
}
```

You can use the CallDetective Factory like this:

```
// ...
CallDetective cd = CallDetective.Factory.makeCallDetective();
System.out.println(cd.findCaller(0));
```

A slightly adapted version of our old StackTrace class is shown here:

```
/**
 * This is a pre-JDK 1.4 version of the CallDetective.
 * See TJSN 4th edition.
 * http://www.javaspecialists.co.za/archive/Issue004.html
 */
import java.io.*;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
public class CallDetective1_3 implements CallDetective {
  private final Throwable tracer = new Throwable();
  private final StringWriter sw = new StringWriter(1024);
  private final PrintWriter out = new PrintWriter(sw, false);

  public String findCaller(int depth) {
    if(depth < 0) {
      throw new IllegalArgumentException();
    }

    int lineOfInterest = depth + 3;
    sw.getBuffer().setLength(0); // set the buffer back to zero
    tracer.fillInStackTrace();
    tracer.printStackTrace(out);

    LineNumberReader in = new LineNumberReader(
      new StringReader(sw.toString()));

    try {
      String result;
      while((result = in.readLine()) != null) {
        if (in.getLineNumber() == lineOfInterest) {
          return beautify(result);
        }
      }
    } catch(IOException ex) {
      // this should REALLY never happen
      throw new RuntimeException(ex.toString());
    }
    throw new IllegalArgumentException();
  }

  private static String beautify(String raw) {
    raw = raw.trim();
    if (raw.startsWith("at ")) {
      return raw.substring(3);
    }
    return raw;
  }
}
```

The version of JDK 1.4.x onwards is much easier:

```
/**
 * This class is used to determine who called us.
 * It is deliberately not thread-safe.
 */
public class CallDetective1_4 implements CallDetective {
  private final Throwable tracer = new Throwable();

  public String findCaller(int depth) {
    if (depth < 0) {
      throw new IllegalArgumentException();
    }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    tracer.fillInStackTrace();
    return tracer.getStackTrace()[depth+1].toString();
  }
}
```

This could be used as part of a Log class, but I must warn you that it will obviously impact the performance of your application if you overdo it...

```
public class Log {
  private static CallDetective detective =
    CallDetective.Factory.makeCallDetective();
  public static void it(String msg) {
    String source = detective.findCaller(1); //caller of it()
    System.out.println(source + " : " + msg);
  }
}
```

Let's look at some test code:

```
public class Test {
  private CallDetective saps =
    CallDetective.Factory.makeCallDetective();
  {
    System.out.println("Using " + saps.getClass().getName());
  }
  public void f() {
    g();
  }
  public void g() {
    Log.it("where am I now?");
    System.out.println(saps.findCaller(0));
    System.out.println(saps.findCaller(1));
    System.out.println(saps.findCaller(2));
    System.out.println();
    System.out.println();
  }
  public static void main(String[] args) {
    new Test().f();
  }
}
```

I ran this with the three versions of Java that are installed on my notebook:

```
java version "1.3.1_06"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1_06-b01)
Java HotSpot(TM) Client VM (build 1.3.1_06-b01, mixed mode)

Using CallDetective1_3
Test.g(Test.java:11) : where am I now?
Test.g(Test.java:12)
Test.f(Test.java:8)
Test.main(Test.java:20)
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
java version "1.4.0_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.0_01-b03)
Java HotSpot(TM) Client VM (build 1.4.0_01-b03, mixed mode)

Using CallDetective1_4
Test.g(Test.java:11) : where am I now?
Test.g(Test.java:12)
Test.f(Test.java:8)
Test.main(Test.java:20)


java version "1.4.1_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.1_01-b01)
Java HotSpot(TM) Client VM (build 1.4.1_01-b01, mixed mode)

Using CallDetective1_4
Test.g(Test.java:11) : where am I now?
Test.g(Test.java:12)
Test.f(Test.java:8)
Test.main(Test.java:20)
```

And now for the real reason why you should use this...

My grandfather kept on repeating himself: "It's all about the money. It's all about the money. Money makes the world go round. Money is everything. It's all about the money. It's all about the money." At age sixteen, I was not going to listen to anyone anyway, so his efforts were wasted on me, I still don't agree with him ;-)

However, in the Java world, it *is* all about the money. The money here is the currency of performance. By the time I publish the next newsletter, I hope to meet in person one of the authorities on Java Performance Tuning, Jack Shirazi. I want to review his book Java Performance Tuning, 2nd Edition, so stay tuned ... In the meantime, let us look at the performance of the two approaches:

```
public class PerfTest {
  private final CallDetective cd;
  public PerfTest(CallDetective cd) {
    this.cd = cd;
  }
  public void test() {
    f(5);
    f(10);
    f(20);
  }
  public void f(int depth) {
    // build up a big call stack...
    if (depth > 0) {
      f(depth-1);
    } else {
      long time = -System.currentTimeMillis();
      for (int i=0; i<10000; i++) {
        cd.findCaller(0);
      }
      time += System.currentTimeMillis();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
      System.out.println(time + "ms");
    }
  }
  public static void main(String[] args) {
    System.out.println("JDK 1.3 approach to find caller:");
    new PerfTest(new CallDetective1_3()).test();
    System.out.println("JDK 1.4 approach to find caller:");
    new PerfTest(new CallDetective1_4()).test();
    // now let's do it again...
    System.out.println("JDK 1.3 approach to find caller:");
    new PerfTest(new CallDetective1_3()).test();
    System.out.println("JDK 1.4 approach to find caller:");
    new PerfTest(new CallDetective1_4()).test();
  }
}
```

From the results, we can see that the new JDK 1.4 approach is on average 2.7 times faster than the old way of tracing through the stack trace. Please see that in perspective - the biggest speedup will be achieved if you don't dig through the stack trace at all!

```
JDK 1.3 approach to find caller:
1502ms
1923ms
3215ms
JDK 1.4 approach to find caller:
501ms
711ms
1101ms
JDK 1.3 approach to find caller:
1442ms
1903ms
2934ms
JDK 1.4 approach to find caller:
501ms
711ms
1102ms
```

That's if for this week. I have to run downstairs now to send this newsletter off to you so that you have something to think about until the review of the book Java Performance Tuning...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 064] - Disassembling Java Classes

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 64th edition of **The Java(tm) Specialists' Newsletter** sent to 5869 Java Specialists in 94 countries.

I stepped off the plane last Tuesday, having come via London from Estonia, and was hit in the face by 31 degree Celsius heat. I was still wearing my warm clothes that were necessary in the -17 degree Celsius in Estonia (travel report to follow), but very quickly, I adapted again to shorts and T-shirt.

While I was in London, I had the opportunity to meet Jack Shirazi, the author of the best Java performance tuning website. Basically, when I don't know or understand something about Java performance, I go look at the website. Still coming is a review of Jack's book.

Disassembling Java Classes

A few months ago, on our local Java User Group discussion forum, I casually asked the question what was faster: i++, ++i or i+=1. I also mentioned that the answer might surprise them. Wanting to encourage thinking, I never gave the answer ;-)

Yesterday, one of my readers, a bright young student at the Cape Town Technical University, sent me an answer based on a thorough investigation through microbenchmarks. His conclusion was correct, but the approach to getting to the conclusion led to a lot of effort being spent on his part.

The easiest way to decide which is fastest is actually to *disassemble* the Java Class. Note that I am saying disassemble, not decompile. Let's look at the following class:

```
public class Increment {
  public int preIncrement() {
    int i = 0;
    ++i;
    return i;
  }
  public int postIncrement() {
    int i = 0;
    i++;
    return i;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }
  public int negative() {
    int i = 0;
    i-=-1;
    return i;
  }
  public int plusEquals() {
    int i = 0;
    i += 1;
    return i;
  }
}
```

I may ask you: which method is the fastest, which is the slowest? C programmers would probably say that the fastest are i++ and ++i. When you try running these methods many times, you will notice slight differences between them, based on which you run first, how soon the hotspot kicks in, whether you use client or server hotspot, whether you are busy playing an MP3 on your machine while running the test and the phase of the moon.

Instead of measuring the performance, why not investigate what the Java compiler did? You can disassemble a class with the standard javap tool available in your JAVA_HOME\bin directory:

```
javap -c Increment
```

Note that the class must already be compiled. The result is the following:

```
Compiled from Increment.java
public class Increment extends java.lang.Object {
   public Increment();
    public int preIncrement();
    public int postIncrement();
    public int negative();
    public int plusEquals();
}

Method Increment()
   0 aload_0
   1 invokespecial #9 <Method java.lang.Object()>
   4 return

Method int preIncrement()
   0 iconst_0
   1 istore_1
   2 iinc 1 1
   5 iload_1
   6 ireturn

Method int postIncrement()
   0 iconst_0
   1 istore_1
   2 iinc 1 1
   5 iload_1
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
   6 ireturn

Method int negative()
   0 iconst_0
   1 istore_1
   2 iinc 1 1
   5 iload_1
   6 ireturn

Method int plusEquals()
   0 iconst_0
   1 istore_1
   2 iinc 1 1
   5 iload_1
   6 ireturn
```

Now when we look at the different methods of incrementing the local variable, we can see that they are actually identical!

Next time you are thinking of measuring performance using System.currentTimeMillis(), think of also looking at the generated byte code. It might save you a lot of time.

I want to thank Dani•l Maree for inspiring me to write this newsletter through his thorough research as to which increment method was faster.

Kind regards

Heinz

P.S. I hope you all enjoy Valentine's day. Remember to leave work early today - i.e. before 8pm ;-)

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 065] - Wait, Cursor, Wait!

Author: Herman Lintvelt (Polymorph Systems)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 65th edition of **The Java(tm) Specialists' Newsletter**, sent to 6077 Java Experts in almost 100 countries. This week we are privileged that my good friend Herman Lintvelt is again showing us some tricks. This newsletter is on how to handle the wait cursor and modal dialogs. Thanks Herman!

Next week I will again be in Germany, and one of the things I will be doing is an in-house Design Patterns Course at a company in Frankfurt. The timing of this course is fortunate, in that it enables me to visit my dear Grandmother for her 90th birthday as well. A special "thank you" to all those who have made it possible to coordinate the dates. If you are in Frankfurt and you would like to meet up for a beer with some fellow Java programmers on the 12th of March 2003, please send me an email.

Herman Lintvelt is one of very few internationally approved trainers to present our Maximum Solutions courses on Design Patterns and Java.

I recently received a question by Michael Ambrose (forwarded to me by Heinz) about wait cursors in Swing, specifically in conjunction with modal dialogs. One of the questions asked was how to set the wait cursor on the parent window/frame of a modal dialog. But wait, we'll come to that.

Everyone who has developed GUI applications, has been faced with the problem of how to handle those lllooooonnggg operations. Especially in the very warm Februaries we have in Worcester [hk: A small village in South Africa, where the tar melts in winter from the heat], operations take very long indeed. This newsletter is not about handling all the issues involved - since that is the stuff of numerous letters - but focus on wait cursors.

Wait Cursor

What is the intent of wait cursors? To tell the user: "Hey, everywhere you see a wait cursor you can't do nothing." Of course having some progress indication while the application is busy with a long operation is also a good idea. (**Law 1 concerning GUIs: The GUI should ALWAYS be responsive, even if it is only indicating how busy the application is.**) I recently discovered a nice class to use for feedback on long operations: javax.swing.ProgressMonitor. However, most of the time we need a wait cursor (also known as an hourglass cursor).

Most of you probably already know how to use wait cursors in Swing, but let me go ahead and give an example of a useful CursorToolkitOne class, implementing an interface for the constants:

```
import java.awt.*;

public interface Cursors {
  Cursor WAIT_CURSOR =
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    Cursor.getPredefinedCursor(Cursor.WAIT_CURSOR);
  Cursor DEFAULT_CURSOR =
    Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR);
}
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/** Basic CursorToolkit that still allows mouseclicks */
public class CursorToolkitOne implements Cursors {
  private CursorToolkitOne() { }

  /** Sets cursor for specified component to Wait cursor */
  public static void startWaitCursor(JComponent component) {
    RootPaneContainer root =
      (RootPaneContainer)component.getTopLevelAncestor();
    root.getGlassPane().setCursor(WAIT_CURSOR);
    root.getGlassPane().setVisible(true);
  }

  /** Sets cursor for specified component to normal cursor */
  public static void stopWaitCursor(JComponent component) {
    RootPaneContainer root =
      (RootPaneContainer)component.getTopLevelAncestor();
    root.getGlassPane().setCursor(DEFAULT_CURSOR);
    root.getGlassPane().setVisible(false);
  }

  public static void main(String[] args) {
    final JFrame frame =  new JFrame("Test App");
    frame.getContentPane().add(
      new JLabel("I'm a Frame"), BorderLayout.NORTH);
    frame.getContentPane().add(
      new JButton(new AbstractAction("Wait Cursor") {
        public void actionPerformed(ActionEvent event) {
          System.out.println("Setting Wait cursor on frame");
          startWaitCursor(frame.getRootPane());
        }
      }));
    frame.setSize(800, 600);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.show();
  }
}
```

CursorToolkitOne only has two class methods: startWaitCursor(...) and stopWaitCursor(...). You pass a JComponent as parameter, and each method finds the RootPaneContainer (i.e. the uppermost container that contains the component) and then sets the Cursor of the GlassPane of this container to either the default or the wait cursor. It then sets this GlassPane's visibility to true or false. As easy as pie, or is it? What happens if we run the main method?

A JFrame is displayed, with a label and a single button. Pressing the button results in the wait cursor being set on the frame via the startWaitCursor method. This part is

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

still fine, but what happens if you click on the button again? Unfortunately the button's action is performed again (you can see the extra System.out.println). So we have a wait cursor, but is does not actually stop the input.

So let's upgrade to CursorToolkitTwo:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/** Basic CursorToolkit that swallows mouseclicks */
public class CursorToolkitTwo implements Cursors {
  private final static MouseAdapter mouseAdapter =
    new MouseAdapter() {};

  private CursorToolkitTwo() {}

  /** Sets cursor for specified component to Wait cursor */
  public static void startWaitCursor(JComponent component) {
    RootPaneContainer root =
      ((RootPaneContainer) component.getTopLevelAncestor());
    root.getGlassPane().setCursor(WAIT_CURSOR);
    root.getGlassPane().addMouseListener(mouseAdapter);
    root.getGlassPane().setVisible(true);
  }

  /** Sets cursor for specified component to normal cursor */
  public static void stopWaitCursor(JComponent component) {
    RootPaneContainer root =
      ((RootPaneContainer) component.getTopLevelAncestor());
    root.getGlassPane().setCursor(DEFAULT_CURSOR);
    root.getGlassPane().removeMouseListener(mouseAdapter);
    root.getGlassPane().setVisible(false);
  }

  public static void main(String[] args) {
    final JFrame frame = new JFrame("Test App");
    frame.getContentPane().add(
      new JLabel("I'm a Frame"), BorderLayout.NORTH);
    frame.getContentPane().add(
      new JButton(new AbstractAction("Wait Cursor") {
        public void actionPerformed(ActionEvent event) {
          System.out.println("Setting Wait cursor on frame");
          startWaitCursor(frame.getRootPane());
        }
      }));
    frame.setSize(800, 600);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.show();
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

We added a MouseAdapter that does nothing to the GlassPane and this prevents any MouseEvents from getting through to the underlying components. Why does it work this way? Well, that's a topic for another discussion.

Wait Cursors and Modal dialogs...

However, the original question was not only about wait cursors, but their use with modal dialogs, and specifically the parent frame or window of the modal dialog.

The intent of a modal dialog is to deny access to any other part of the application GUI, while retaining access to the dialog. So what happens if the dialog initiates a background operation that takes long to execute (**GUI Law 2: never execute a potential long operation from the main GUI thread**), and you need to indicate with a wait cursor that the dialog GUI is off limits for the moment? Easy: use CursorToolkitTwo.startWaitCursor to set the wait cursor on the dialog.

And then one day when a client is playing with his mouse while waiting for the wait cursor to disappear (since there is no progress indication because of tight deadlines), he sees that the cursor changes back to the default cursor when he moves the mouse out of the dialog unto the main frame of the application. I can already see the Problem Report: "No wait cursor is shown while the application is busy; only the dialog has a wait cursor, but mouse-clicks have no effect even though there is no wait cursor."

How can we fix this gross enfringement of human rights?

Solution 1

Hey, this should be simple: I only have to get the parent of the modal dialog (in most cases a JFrame), and set the wait cursor on it:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/** First attempt at a solution */
public class SolutionOne {
  private static JDialog createDialog(final JFrame frame) {
    final JDialog dialog = new JDialog(frame, "I'm Modal", true);
    dialog.getContentPane().add(
      new JLabel("I'm a busy modal dialog"));
    dialog.getContentPane().add(
      new JButton(new AbstractAction("Wait Cursor") {
        public void actionPerformed(ActionEvent event) {
          setWaitCursor(dialog);
        }
      }));
    dialog.setSize(300, 200);
    return dialog;
  }

  public static void setWaitCursor(JDialog dialog) {
    System.out.println("Setting Wait cursor on frame");
    CursorToolkitTwo.startWaitCursor(
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      ((JFrame)dialog.getOwner()).getRootPane());
    System.out.println("Setting Wait cursor on dialog");
    CursorToolkitTwo.startWaitCursor(dialog.getRootPane());
  }

  public static void main(String[] args) {
    final JFrame frame =  new JFrame("Solution One");
    frame.getContentPane().add(
      new JLabel("I'm a Frame"), BorderLayout.NORTH);
    frame.getContentPane().add(
      new JButton(new AbstractAction("Show Dialog") {
        public void actionPerformed(ActionEvent event) {
          System.out.println("Showing dialog");
          createDialog(frame).show();
        }
      }));
    frame.setSize(800, 600);
    frame.setVisible(true);
  }
}
```

I'm not going to discuss all the Swing code; basically a JFrame is created that contains a button. When pressed, this button will show a dialog that contains a button. If this dialog button is pressed, then the setWaitCursors method will be called, which attempts to set the wait cursor on both the dialog and it's parent frame by using our CursorToolkitTwo.

Run it. Press the buttons. It doesn't work :-(. Yes, the wait cursor is set on the dialog, but not on the frame behind it.

Why not?! Well, as soon as a modal dialog is displayed, the current AWT event pump (the mechanism that handles mouse, keyboard and other events) is blocked, and a new event pump is started. As soon as the modal dialog is closed, the previous event pump is unblocked. This means that if the modal dialog in SolutionOne is closed, the wait cursor will suddenly be set on the frame. "Betterlate than never" they say, but this is an example of "better never than late" :-)

Solution 2

There are a few ways around this. If you have access to the code that calls the dialog, but not to the dialog code, you can try to first set the wait cursor on the dialog's parent (the JFrame in our example) before displaying the dialog. (Maybe this is something that should happen by default when displaying a modal dialog? Interesting topic for a discussion; I'd be interested in what you say about this. Email me at Herman to voice you're opinions.)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 * This is a second attempt - but native code changes the
 * cursor back to the default cursor.
 */
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public class SolutionTwo {
  private static JDialog createDialog(final JFrame frame) {
    final JDialog dialog = new JDialog(frame, "I'm Modal", true);
    dialog.getContentPane().add(
      new JLabel("I'm a busy modal dialog"));
    dialog.getContentPane().add(
      new JButton(new AbstractAction("Wait Cursor"){
        public void actionPerformed(ActionEvent event) {
          setWaitCursor(dialog);
        }
      }));
    dialog.setSize(300, 200);
    return dialog;
  }

  private static void setWaitCursor(final JDialog dialog) {
    System.out.println("Setting Wait cursor on dialog");
    CursorToolkitTwo.startWaitCursor(dialog.getRootPane());
  }

  public static void main(String[] args) {
    final JFrame frame = new JFrame("Solution Two");
    frame.getContentPane().add(
      new JLabel("I'm a Frame"), BorderLayout.NORTH);
    frame.getContentPane().add(
      new JButton(new AbstractAction("Show Dialog") {
        public void actionPerformed(ActionEvent event){
          System.out.println("Setting Wait cursor on frame");
          CursorToolkitTwo.startWaitCursor(frame.getRootPane());
          System.out.println("Showing dialog");
          createDialog(frame).show();
        }
      }));
    frame.setSize(800, 600);
    frame.show();
  }
}
```

SolutionTwo is similar to SolutionOne, except for the changes listed above. The setWaitCursor method now only sets the wait cursor on the dialog, but in the showDialogAction.actionPerformed method, we set the wait cursor on the frame before showing the dialog.

Run it.

Another failure :-(

Why does this not work? We've set the wait cursor before the main event pump got blocked, and apparently Swing (or AWT) resets the cursor to the default cursor on the rest of the components (i.e. everything except the modal component). You can check the JDC Bug Parade (bug nr 4282540) for their reasons why this is so.

Solution 3

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Well, the easy way to fix this (if you have access to the dialog code) is to make the dialog non-modal, but then set the wait cursor on the frame before showing the dialog. It prevents access to the frame, while allowing the wait cursor to be set, and also allows the wait cursor to be set on the dialog. You then basically have SolutionTwo, but a non-modal dialog is now created.

Here is a WaitEnabledDialog that automatically sets the JFrame cursor to an hourglass whenever the dialog is opened and resets it to the default cursor when the dialog closes:

```
import java.awt.event.*;
import javax.swing.*;

public class WaitEnabledDialog extends JDialog {
  public WaitEnabledDialog(final JFrame owner, String title) {
    super(owner, title, false);
    addWindowListener(new WindowAdapter() {
      public void windowOpened(WindowEvent e) {
        CursorToolkitTwo.startWaitCursor(owner.getRootPane());
      }
      public void windowClosing(WindowEvent e) {
        CursorToolkitTwo.stopWaitCursor(owner.getRootPane());
      }
    });
  }
}
```

Our third solution now looks like so:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 * Here is a solution where we make the modal dialog non-modal.
 * Since we disable mouse clicks on the frame, it is actually
 * the same as a modal dialog.
 */
public class SolutionThree {
  private static JDialog createDialog(final JFrame frame) {
    final WaitEnabledDialog dialog =
      new WaitEnabledDialog(frame, "I'm not Modal");
    dialog.getContentPane().add(
      new JLabel("I'm a busy non-modal dialog"));
    dialog.getContentPane().add(
      new JButton(new AbstractAction("Wait Cursor") {
        public void actionPerformed(ActionEvent event){
          setWaitCursor(dialog);
        }
      }));
    dialog.setSize(300, 200);
    return dialog;
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public static void setWaitCursor(final JDialog dialog) {
    System.out.println("Setting Wait cursor on dialog");
    CursorToolkitTwo.startWaitCursor(dialog.getRootPane());
  }

  public static void main(String[] args) {
    final JFrame frame =  new JFrame("Solution Three");
    frame.getContentPane().add(
      new JLabel("I'm a Frame"), BorderLayout.NORTH);
    frame.getContentPane().add(
      new JButton(new AbstractAction("Show Dialog") {
        public void actionPerformed(ActionEvent event){
          JDialog dialog = createDialog(frame);
          System.out.println("Showing dialog");
          dialog.show();
        }
      }));
    frame.setSize(800, 600);
    frame.show();
  }
}
```

Semi-modal Dialogs

Talking of non-modal dialogs, I came across the idea of "semi-modal" dialogs on the Web. Basically it's a less strict version of a modal dialog, in that certain specified components of the parent frame/window are still accessible, even though the semi-modal dialog is displayed. I did not fully agree with the situation it was used in: basically the author wanted to give users the option of cancelling the current operation (the semi-modal dialog gets input parameters from the user) by selecting another function on the frame's toolbar. Why not just add a "Cancel" button to the dialog? (**GUI Law 3: the GUI should be as simple and intuitive as possible**). I actually cannot think of any scenario where you would want to use a semi-modal dialog, however that might be because I wanted to use it for our wait cursor problem, but could not find a way in which it will be easier to use than **Solution 3**.

I must admit, though, the idea of a semi-modal dialog is an interesting one. You can check out the code as well as a discussion of a class called Blocker at JavaWorld. Blocker extends the java.awt.EventQueue class that handles the queueing and dispatching of AWT events. It allows one to register components that should be "blockable", and then you can enable or disable the blocking (i.e. switch between semi-modal and normal mode).

Threading solutions

Aha! Why not use multiple threads and trick AWT into keeping the wait cursor on the frame, while showing a modal dialog? Because it's a very bad idea to have more than one event pump working, and even if you don't call dialog.setVisible(true) from the main Swing/AWT thread, it will still block the main EventDispatchThread. You can update SolutionTwo to set the wait cursor on the frame, and then start another thread that sleeps a few hundred milliseconds before displaying the dialog. The wait cursor will be visible on the frame while the specified number of milliseconds tick off, and then behold: it is once again a default cursor just as the dialog appears.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

And the Keyboard?

You probably noticed that I did not mention keyboard input at all. Well, the above solutions only prevent mouse input. Go have a look at KeyboardFocusManager as an exercise (if you're using JDK 1.4+).

Happy waiting until next time :-)

Herman

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 066] - Book Review: Java Performance Tuning by Jack Shirazi

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 66th edition of **The Java(tm) Specialists' Newsletter**, sent to 6138 Java Experts in 94 countries. The newslist server is going to start throwing out dead email addresses, so the side effect is that the number of subscribers will likely decrease. If you did not receive this email, please respond by replying to this email and telling me that you did not receive it.

Last week I had great fun with some programmers in Frankfurt. We did the Design Patterns Course and good laughs together and of course we all learned new things. Thanks especially to my friend Carl Smotricz (who used to host our newsletter archive) for organising this event. One of the highlights of my trip was meeting with some of our newsletter subscribers for a good German beer.

At the end of April, we are running another Design Patterns Course in Cape Town. See the end of this email for more information. This course is aimed mainly at South Africans, but we welcome you from anywhere in the world. Please let us know as soon as possible if you would like to attend - the course is already filling up ...

Meeting Jack Shirazi

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

During my recent travels, I had the opportunity to meet with Jack Shirazi, famous as the author of the landmark book Java Performance Tuning. Besides consulting on performance related issues, Jack also maintains an excellent website, which is the #1 resource for expert Java programmers who want to keep up to date with Java performance. I particularly like his website because true to the topic, it is fast to navigate. The website does not look particularly pretty, but you can easily get to the information you want. There are no frames, no pictures, no special fonts, but gosh, does it render quickly! When I chatted to Jack about this, he told me that according to the principles in his book, the fastest content that you could possibly serve is static content, i.e. content that is not dynamically generated. Jack therefore generates his entire website and uploads the differences. This works because the content does not change all the time, it would be a bit difficult to build an entire internet banking site like this. However, even the internet banking site should be serving static content whenever this is possible.

Jack and I spent a good few hours chatting about all sorts of topics, from the state of consulting to business ideas, to how many kids we have (at this point I knew that Jack was hopelessly superior at a 3:1 ratio and we agreed to decide on greatness by means of an arm-wrestle). The waitress was quite amused since neither of us really had any idea of what we were doing. There are two types of people in this world: those that can program, and those that can arm-wrestle. Our inability to achieve neither victory nor defeat has placed us firmly into the camp of programmers :-)

Ah, here is a good job interview question: "Hi, so you want to work for the XYZ company where we build important software. What did you think of the {rugby|cricket|soccer|football|basketball|baseball} game on Saturday?" An answer like "Was there a match? Who played?" will get you the job as programmer without any further questions, but saying "Yeah, Ted performed well - he has recovered well since his knee operation that he had in high school - he went to Osborne High from 1984-1989 where he was Valedictorian, did you know?" will only get you a position in the sales department. Mind you, the sales department is not a bad place to be - at least you can watch sport on the weekend and tell your boss and spouse that you are doing important work.

Let us not get sidetracked...

Java Performance Tuning 2nd Edition

Naturally I did not just travel 10000km to speak to Jack about kids and lucrative consulting jobs. [To those of you who do not have kids, once you have them, you will realise that they are far more important than anything else in life, even Java.]

The real reason I met with Jack was to speak about the 2nd Edition of his book Java Performance Tuning. When Jack wrote the first edition of his book, there was not much else available. The one book that I have mentioned in this newsletter is Java 2 Performance and Idiom Guide, and that was available before Jack's. However, Jack's book focuses far more on performance than the other book. In addition, Jack adds value to the reader with his informative website that is worthwhile to explore.

The book contains chapters on the following performance topics:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Chapter 1 gives general guidelines on how to tune. It will help you to formulate a good tuning strategy following a methodical process.

Chapter 2 covers the tools you need to use while tuning.

Chapter 3 looks at the SDK, including Virtual Machines (VMs) and compilers. Jack does not cover JDK 1.4.1, only up to JDK 1.4.0. This is understandable, because at the time of writing, JDK 1.4.1 had not been released.

Chapters 4 - 12 cover various techniques you can apply to Java code.

Chapter 12 looks at tuning techniques specific to distributed applications.

Chapter 13 steps back from the low-level code-tuning techniques examined throughout most of the book and considers tuning at all other stages of the development process.

Chapter 14 is a quick look at some operating system-level tuning techniques.

Chapters 15 - 18 are new in this edition, providing the information you need to tune J2EE applications. Chapter 15 describes tuning considerations that are common to all J2EE applications. Chapter 16 looks at specifics for tuning JDBC, including optimizing transactions and SQL, while Chapter 17 provides important information for speeding up servlets and JSPs. Chapter 18 describes performance considerations for Enterprise JavaBeans (EJBs).

Finally, **Chapter 19** covers a wide range of additional resources where you can learn more about Java performance tuning. Chapter 1: Introduction

The book starts with a chapter outlining a performance tuning strategy that you should follow when attempting to tune an application. Follow it, and you will be successful. Ignore it, and you will forever be tuning and never get any acknowledgements for your achievements. At the end of each chapter is a performance checklist of the major points of that chapter. For example:

Specify the required performance.

Make your benchmarks long enough: over five seconds is a good target.

Break down distributed application measurements into components, transfer layers, and network transfer times.

Tune systematically: understand what affects the performance; define targets; tune; monitor and redefine targets when necessary.

Work with user expectations to provide the appearance of better performance.

Quality-test the application after any optimizations have been made.

Document optimizations fully in the code. Retain old code in comments. Chapter 6: Exceptions, Assertions, Casts, and Variables

Many moons ago, when Java had just been invented, programmers invented strange ways to write optimal code. One of the weirdest abominations was the logic that a **try**/**catch** was for free when no exception was thrown and that it was cheaper than **if** and **instanceof** statements. For example:

```
public class TryCatch {
  public static boolean test1(Object o) {
    try {
      Integer i = (Integer)o;
      return false;
    } catch(Exception e) {
      return true;
    }
  }

  public static boolean test2(Object o) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    if (o instanceof Integer) {
      Integer i = (Integer)o;
      return false;
    } else {
      return true;
    }
  }
}
```

The thinking was that a try block was free, so therefore if you almost always have Integer objects, it will be faster to just cast and hope for the best, than to first test if it really is an Integer. test1() should therefore be faster than test2() most of the time, unless we passed in an object that was not an Integer.

But is it really faster? Let us have a look:

```
public class TryCatchCostTest extends TryCatch {
  public static void main(String[] args) {
    Integer i = new Integer(3);
    Boolean b = new Boolean(true);
    long TEST_DURATION = 2 * 1000;
    boolean res;
    long stop;

    stop = TEST_DURATION + System.currentTimeMillis();
    int test1_i = 0;
    // we do not want to test with every increment, otherwise the
    // call to System.currentTimeMillis() will dwarf the rest of
    // our calls.
    while(((test1_i % 1000) != 0)
        || (System.currentTimeMillis() < stop)) {
      test1_i++;
      res = test1(i);
    }
    System.out.println("test1(i) executed " + test1_i + " times");

    stop = TEST_DURATION + System.currentTimeMillis();
    int test1_b = 0;
    while(((test1_b % 1000) != 0)
        || (System.currentTimeMillis() < stop)) {
      test1_b++;
      res = test1(b);
    }
    System.out.println("test1(b) executed " + test1_b + " times");
    System.out.println("test1(i) was " + (test1_i / test1_b)
                + " times faster");

    stop = TEST_DURATION + System.currentTimeMillis();
    int test2_i = 0;
    while(((test2_i % 1000) != 0)
        || (System.currentTimeMillis() < stop)) {
      test2_i++;
      res = test2(i);
    }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    System.out.println("test2(i) executed " + test2_i +  " times");

    stop = TEST_DURATION + System.currentTimeMillis();
    int test2_b = 0;
    while(((test2_b % 1000) != 0)
       || (System.currentTimeMillis() < stop)) {
      test2_b++;
      res = test2(b);
    }
    System.out.println("test2(b) executed " + test2_b +  " times");
  }
}
```

When I run this test on JDK 1.4.1_01 with my Pentium 4m 1.7GHz, I get the following values under client hotspot:

test1(i) executed 22700000 times
test1(b) executed 126000 times
test1(i) was 180 times faster
test2(i) executed 52420000 times
test2(b) executed 53015000 times

Oops, it appears that logic was incorrect! test2() is about twice as fast as the best test1()! How does this look with the server hotspot compiler?

java -server TryCatchCostTest
test1(i) executed 50310000 times
test1(b) executed 43000 times
test1(i) was 1170 times faster
test2(i) executed 51915000 times
test2(b) executed 51952000 times

The difference between test1() and test2() is not as pronounced as with the client hotspot, but test2() is still marginally faster. However, the version of test1() that causes the exception has become three times slower!

Under what circumstances did this idea proliferate? My suspicion is that it used to be true for the interpreted version of Java, probably around JDK 1.0. Have a look at what happens when we run this test with the -Xint switch. It appears that test1(i) is indeed faster!

java -Xint TryCatchCostTest
test1(i) executed 19153000 times
test1(b) executed 180000 times
test1(i) was 106 times faster
test2(i) executed 14522000 times
test2(b) executed 18719000 times

The book is great in that for all important statistics, it compares performance results for Sun JDK 1.1.8, 1.2.2, 1.3.1, 1.3.1 -server, 1.4.0, 1.4.0 -server and 1.4.0 -Xint. This is probably the most useful part of the book, and also lends credibility to the statements that Jack makes. If Jack makes a statement without proof, you cannot assume that it is true. However, when the statement is backed up with values, you can be assured that it is probably correct.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Can you paint yourself into a corner?

Kent Beck is famous for his phrase: "Make it run, make it right, make it fast."

Unless we can compile it, we do not need to try tune the performance. Once we get it to compile, we have to make sure that it is correct. What is the point of having a blazingly fast algorithm that is incorrect? I can sort any collection of elements in O(1) if it does not have to be correct.

In my doctoral thesis, I examined how one could analytically evaluate the performance of a distributed system written in the Specification and Discription Language (SDL). Since there were few analytical techniques for performance evaluation of SDL systems, I converted SDL to a new form of Petri net based on the Queuing Petri Net invented by the University of Dortmund. There are many correctness and performance testing techniques for Petri nets, so the first step was to determine whether the net contained deadlocks, was unbounded or had livelocks. Once the correctness had been determined, I could apply the numerous performance techniques to the net and then convert the results back to the original SDL system.

Due to the complexity of the Markov Chain that was generated from the Petri net, we could only analyse small systems, but then, it was a doctoral thesis, not a commercial product. You are allowed to dream a bit in theory when you do a PhD, infact, a PhD is where you investigate everything about nothing.

In the real world, we usually stop after the first test: "does it compile?" Most software in the world has not been tested before being delivered. Why? Because testing that it works correctly is a tedious job that takes time and programmers do not make mistakes anyway. I am yet to walk into a company and ask whether they have unit tests, without getting a nervous reaction. When I then ask what they are doing about performance I get an even more nervous reaction.

When the system then performs badly, they have to call in trouble-shooters like Jack and myself to tune their applications. Performance involves both CPU times and memory sizes. A few newsletter ago, I casually asked whether you wanted to know how to detect OutOfMemoryErrors ahead of time. More than 100 readers responded that they were in desperate need of such a tool. Jack and I do not mind this state of affairs at all - that is a good way to earn our keep. However, you will have a problem when you need Jack and me but you cannot afford our rates. Then you will kick yourself for putting performance last.

I asked Jack about the philosophy of "Make it run, make it right, make it fast" and Jack mentioned to me that there are situations where this does not work. There are ways in which you can paint yourself in a corner so that the only alternative is to throw away the application and start again. This piqued my curiosity, and I asked Jack for an example.

"An example of painting yourself in the corner is when you do not use the Page-by-Page Iterator Design Pattern from the beginning"

As I thought about the troubled projects I had consulted for, I realised that Jack was correct. You can get a better optimising compiler, you can reduce the number of Strings that get created, you can change the collection you use, but if you do not have the Page-by-Page Iterator Design Pattern in your design from the beginning,

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

you will be p*ssing against the wind. This applies for 3-tier and for 2-tier applications. The P-b-P Iterator can be added to your system after-the-fact, but it will require plenty of work and cause errors.

Conclusion

Should you purchase the Java Performance Tuning Book? There is no easy answer to that question. If you do not have the US$ 44.95 to pay for the book, it will be unlikely that you will be able to afford to pay for Jack or me to help you tune your system. Basically, you would be on your own. You can also browse Jack's website or look at our newsletter archive. Otherwise, if you are ***serious*** about writing good Java code that is well-tuned, you would do well to purchase a copy of Jack's book, and read through it over a period of a few months.

That is all I have to say on this topic. The book could be converted into about 30 newsletters, that is how much information is contained therein.

Until next month - and I do hope that some of you will come to our Design Patterns Course in Cape Town from the 28th to 30th of April 2003.

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 067] - BASIC Java

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 67th edition of **The Java(tm) Specialists' Newsletter**, sent to 6299 Java Specialists in 94 countries. I am in the process of printing a softcover book containing all the newsletters up to and including the 65th issue. The book is primarily for students on my courses, but if you would like a copy at ZAR 295, please send me an email. A softcover book is practical since you can read it while lying on the beach.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

In the western world, we celebrate the 1st of April with jokes, trying to catch out unsuspecting and gullible people. Last year, I sent out a newsletter in which I announced that from that day forth, you would have to pay to become unsubscribed, and that the unsubscription fee would automatically be deducted off your credit card. Naturally, quite a few subscribers sent me emails such as: "haha, look at the date". Others sent me outraged emails telling me that they had not agreed to this and that I could not do it. My wife Helene had warned me against putting in the joke, but being a stubborn South African German, I naturally did not listen. This year, I was going to repeat my folly, but having thought about it for a while, have decided that with a hundred countries on my list, an April fool's newsletter would only serve to confuse a significant number of readers.

HOWEVER, in this newsletter I have some lighthearted material on how you can write BASIC code in Java. In the middle of March I was in Germany visiting my grandmother for her 90th birthday, and while I was there, managed to present a Design Patterns Course to a company in Frankfurt. Our friend Carl Smotricz, who used to host our newsletter on his website, was one of the 13 course participants. While Carl was taking me to the train station after the second day of lectures, Carl told me that it was possible to program GOTO in Java. Naturally I was curious, so I asked Carl to give me an example. The theme sounded like an ideal opportunity for an April 1st joke, so I will send this newsletter to you with the warning that Carl and I were basically having fun and seeing how far we could push the Java Programming Language to look like **BASIC**.

BASIC Java

Carl and I have something in common. Both of us started programming in BASIC. **BASIC** stands for **B**eginners **A**ll-purpose **S**ymbolic **I**nstruction **C**ode and is the easiest language in the world to understand. It is not surprising that Visual **BASIC** is so popular. We even have plans to produce a Design Patterns Course for Visual Basic .NET (the Delphi version is already underway - and that is not an April Fools joke ;-)

BASIC has the following advantages over Java:

BASIC is for **beginners**. Most programmers in the world today have less than 10 years experience and therefore should be considered as beginners. Java is definitely not for beginners. It is for expert programmers. It gives you more ways of shooting yourself in the foot than an AK47. (I do not consider VB.NET to be BASIC, but rather just another flavour of Java.)
BASIC allows global variables. Global variables reduce the number of instances of variables needed, they reduce the namespace. Global variables make it easy to *write* a program, although maintenance becomes almost impossible. Do not you believe me that global variables make coding easier? Think of a Design Pattern. Think of *any* Design Pattern. Any one. Name it out loud, and then please go to the end of this email and look at the P.S.
BASIC supports the GOTO statement. GOTO is perhaps not the darling of Edsger W. Dijkstra but it does allow us to write *interesting* succinct code, which certainly ensures that we will be employed in a maintenance role for many years to come.
BASIC is widely understood and supported. Say "goodbye" to expensive finder fees from recruitment agencies. Hire one of thousands of BASIC programmers and pay them very little. Or, since BASIC is for beginners, take someone straight out of arts to do the programming for you.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Consider the following **BASIC** program. Don't just skip over it, try to understand what is happening. If you read it carefully, you will see that it constructs a list of 10 numbers, then sorts them using the Bubble Sort algorithm. (Thanks Carl for this piece of code.)

```
10 DIM N(10):RANDOMIZE TIMER
20 I = 0
30 N(I) = INT(1000 * RND)
40 I = I + 1: IF I < 10 GOTO 30
50 S% = 1 : I = 0
60 IF N(I) < N(I + 1) GOTO 80
70 T = N(I) : N(I) = N(I + 1) : N(I + 1) = T : S% = 0
80 I = I + 1 : IF I < 9 GOTO 60
90 IF S% = 0 GOTO 50
100 I = 0
110 PRINT N(I)
120 I = I + 1 : IF I < 10 GOTO 110
130 STOP
```

When we run this with Microsoft QuickBasic 3.0, we see something like:

```
91
100
460
589
618
622
650
730
888
918
```

If we wanted to convert this to Java, we could either hire an expensive Java programmer (who probably would not understand the above code), or we could write a tool to automatically translate the above code to "Java". Remember the GIGO principle - Garbage in - Garbage out.

Is it possible to translate this to Java? Yessir!

```
public class Bubble extends Basic {
  // Original Code:
  // 10 DIM N(10):RANDOMIZE TIMER
  // 20 I = 0
  // 30 N(I) = INT(1000 * RND)
  // 40 I = I + 1: IF I < 10 GOTO 30
  // 50 S% = 1 : I = 0
  // 60 IF N(I) < N(I + 1) GOTO 80
  // 70 T = N(I) : N(I) = N(I + 1) : N(I + 1) = T : S% = 0
  // 80 I = I + 1 : IF I < 9 GOTO 60
  // 90 IF S% = 0 GOTO 50
  // 100 I = 0
  // 110 PRINT N(I)
  // 120 I = I + 1 : IF I < 10 GOTO 110
  public static void main(String[] args) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    int I=0;int S=0;int N[] = null;
    while(jump != -1) {
     try {
       switch(jump) {
         case 10: N = DIM(10);
         case 20: I = 0;
         case 30: N[I] = INT(1000 * RND());
         case 40: I = I + 1; if (I < 10) GOTO (30);
         case 50: S = 1; I = 0;
         case 60: if( N[I] < N[I + 1]) GOTO (80);
         case 70: int T = N[I] ; N[I] = N[I + 1] ; N[I + 1] = T ; S = 0;
         case 80: I = I + 1 ; if (I < 9) GOTO (60);
         case 90: if (S == 0) GOTO (50);
         case 100: I = 0;
         case 110: PRINT (N[I]);
         case 120: I = I + 1; if (I < 10) GOTO (110);
         case 130: STOP();
       }
       // if there was no GOTO then we want to end the program
       STOP();
     } catch(GotoException ex) {
       // GOTO was called, and a GotoException has caused the
       // control to pass outside of the switch statement
     }
    }
   }
}
```

The output would be something along the lines of:

```
49
102
179
212
351
425
488
877
893
909
```

What does the code for Basic.java look like?

```
public class Basic {
  // GotoException is thrown by GOTO to avoid having to call
  // "break" after the call to GOTO in the switch statement.
  // I warned you before that Exceptions were dangerous ;-)
  protected static final class GotoException
      extends RuntimeException {
    private GotoException() {}
  }
  private static final GotoException gotoEx =
    new GotoException();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
  public static int jump=10;
  public static void GOTO(int line) {
    jump = line;
    throw gotoEx;
  }
  // STOP changes the current line to be -1, which ends the
  // program
  public static void STOP() {
    GOTO (-1);
  }
  public static void PRINT(String s) {
    System.out.println(s);
  }
  public static void PRINT(int i) {
    System.out.println(i);
  }
  public static int[] DIM(int n) {
    return new int[n];
  }
  public static double RND() {
    return Math.random();
  }
  public static int INT(double d) {
    return (int)d;
  }
}
```

What makes me scared is that the code runs and actually works. Perhaps one day Carl and I will write a tool to automatically convert legacy BASIC code to Java .....

Kind regards, and don't get caught out tomorrow!

Heinz

P.S. You probably thought of the Singleton? The Singleton's intent is to provide a global access point to an object. Is that not a global variable?

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 068] - Appending Strings

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 68th edition of **The Java(tm) Specialists' Newsletter**, sent to 6400 Java Specialists in 95 countries. The softcover book of all newsletters up to the 65th issue is now available for purchase within South Africa. Please have a look at our webpage. I keep a copy of the book next to my bed, for obvious reasons (when I cannot wake up in the mornings, I read a few of the newsletters, that wakes up my brain and puts me in a good mood).

Since our last newsletter, we have had two famous Java authors join the ranks of subscribers. It gives me great pleasure to welcome **Mark Grand** and **Bill Venners** to our list of subscribers. Mark is famous for his three volumes of Java Design Patterns books. You will notice that I quote Mark in the brochure of my Design Patterns course. Bill is famous for his book Inside The Java Virtual Machine. Bill also does a lot of work training with Bruce Eckel.

Our last newsletter on BASIC Java produced gasps of disbelief. Some readers told me that they now wanted to unsubscribe, which of course I supported 100%. Others enjoyed it with me. It was meant in humour, as the warnings at the beginning of the newsletter clearly indicated.

For those living in Cape Town, South Africa, we are doing another Design Patterns Course in May 2003, please see the advert at the bottom of this newsletter. If you come on that course, you will also receive a free copy of the Java Specialists' Newsletters Book.

Appending Strings

The first code that I look for when I am asked to find out why some code is slow is concatenation of Strings. When we concatenate Strings with += a whole lot of objects are constructed.

Before we can look at an example, we need to define a Timer class that we will use for measuring performance:

```
/**
 * Class used to measure the time that a task takes to execute.
 * The method "time" prints out how long it took and returns
 * the time.
 */
public class Timer {
  /**
   * This method runs the Runnable and measures how long it takes
   * @param r is the Runnable for the task that we want to measure
   * @return the time it took to execute this task
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
   */
  public static long time(Runnable r) {
    long time = -System.currentTimeMillis();
    r.run();
    time += System.currentTimeMillis();
    System.out.println("Took " + time + "ms");
    return time;
  }
}
```

In the test case, we have three tasks that we want to measure. The first is a simple += String append, which turns out to be extremely slow. The second creates a StringBuffer and calls the append method of StringBuffer. The third method creates the StringBuffer with the correct size and then appends to that. After I have presented the code, I will explain what happens and why.

```
public class StringAppendDiff {
  public static void main(String[] args) {
    System.out.println("String += 10000 additions");
    Timer.time(new Runnable() {
      public void run() {
        String s = "";
        for(int i = 0; i < 10000; i++) {
          s += i;
        }
        // we have to use "s" in some way, otherwise a clever
        // compiler would optimise it away.  Not that I have
        // any such compiler, but just in case ;-)
        System.out.println("Length = " + s.length());
      }
    });

    System.out.println(
       "StringBuffer 300 * 10000 additions initial size wrong");
    Timer.time(new Runnable() {
      public void run() {
        StringBuffer sb = new StringBuffer();
        for(int i = 0; i < (300 * 10000); i++) {
          sb.append(i);
        }
        String s = sb.toString();
        System.out.println("Length = " + s.length());
      }
    });

    System.out.println(
       "StringBuffer 300 * 10000 additions initial size right");
    Timer.time(new Runnable() {
      public void run() {
        StringBuffer sb = new StringBuffer(19888890);
        for(int i = 0; i < (300 * 10000); i++) {
          sb.append(i);
        }
        String s = sb.toString();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      System.out.println("Length = " + s.length());
    }
   });
  }
}
```

This program does use quite a bit of memory, so you should set the maximum old generation heapspace to be quite large, for example 256mb. You can do that with the -Xmx256m flag. When we run this program, we get the following output:

```
String += 10000 additions
Length = 38890
Took 2203ms
StringBuffer 300 * 10000 additions initial size wrong
Length = 19888890
Took 2254ms
StringBuffer 300 * 10000 additions initial size right
Length = 19888890
Took 1562ms
```

You can observe that using StringBuffer directly is about 300 times faster than using +=. Another observation that we can make is that if we set the initial size to be correct, it only takes 1562ms instead of 2254ms. This is because of the way that java.lang.StringBuffer works. When you create a new StringBuffer, it creates a char[] of size 16. When you append, and there is no space left in the char[] then it is doubled in size. This means that if you size it first, you will reduce the number of char[]s that are constructed.

The time that the += String append takes is dependent on the compiler that you use to compile the code. I discovered this accidentally during my Java course last week, and much to my embarrassment, I did not know why this was. If you compile it from within Eclipse, you get the result above, and if you compile it with Sun's javac, you get the output below. I think that Eclipse uses jikes to compile the code, but I am not sure. Perhaps it even has an internal compiler?

```
String += 10000 additions
Length = 38890
Took 7912ms
StringBuffer 300 * 10000 additions initial size wrong
Length = 19888890
Took 2634ms
StringBuffer 300 * 10000 additions initial size right
Length = 19888890
Took 1822ms
```

Why the difference between compilers?

This took some head-scratching, resulting in my fingers being full of wood splinters. I started by writing a class that did the basic String append with +=.

```
public class BasicStringAppend {
  public BasicStringAppend() {
    String s = "";
    for(int i = 0; i < 100; i++) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    s += i;
   }
  }
}
```

When in doubt about what the compiler does, disassemble the classes. Even when I disassembled them, it took a while before I figured out what the difference was and why it was important. The part where they differ is in ***italics***. You can disassemble a class with the tool javap that is in the bin directory of your java installation. Use the -c parameter:

javap -c BasicStringAppend

Compiled with Eclipse:

```
Compiled from BasicStringAppend.java
public class BasicStringAppend extends java.lang.Object {
   public BasicStringAppend();
}

Method BasicStringAppend()
   0 aload_0
   1 invokespecial #9 <Method java.lang.Object()>
   4 ldc #11 <String "">
   6 astore_1
   7 iconst_0
   8 istore_2
   9 goto 34
  12 new #13 <Class java.lang.StringBuffer>
  15 dup
  16 aload_1
  17 invokestatic #19 <Method java.lang.String valueOf(java.lang.Object)>
  20 invokespecial #22 <Method java.lang.StringBuffer(java.lang.String)>
  23 iload_2
  24 invokevirtual #26 <Method java.lang.StringBuffer append(int)>
  27 invokevirtual #30 <Method java.lang.String toString()>
  30 astore_1
  31 iinc 2 1
  34 iload_2
  35 bipush 100
  37 if_icmplt 12
  40 return
```

Compiled with Sun's javac:

```
Compiled from BasicStringAppend.java
public class BasicStringAppend extends java.lang.Object {
   public BasicStringAppend();
}

Method BasicStringAppend()
   0 aload_0
   1 invokespecial #1 <Method java.lang.Object()>
   4 ldc #2 <String "">
   6 astore_1
   7 iconst_0
   8 istore_2
   9 goto 34
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
 12 new #3 <Class java.lang.StringBuffer>
 15 dup
 16 invokespecial #4 <Method java.lang.StringBuffer()>
 19 aload_1
 20 invokevirtual #5 <Method java.lang.StringBuffer
append(java.lang.String)>
 23 iload_2
 24 invokevirtual #6 <Method java.lang.StringBuffer append(int)>
 27 invokevirtual #7 <Method java.lang.String toString()>
 30 astore_1
 31 iinc 2 1
 34 iload_2
 35 bipush 100
 37 if_icmplt 12
 40 return
```

Instead of explaining what every line does (which I hope should not be necessary on a Java **Specialists'** Newsletter) I present the equivalent Java code for both IBM's Eclipse and Sun. The differences, which equate to the disassembled difference, is again in ***italics***:

```
public class IbmBasicStringAppend {
  public IbmBasicStringAppend() {
    String s = "";
    for(int i = 0; i < 100; i++) {
      s = new StringBuffer(String.valueOf(s)).append(i).toString();
    }
  }
}
public class SunBasicStringAppend {
  public SunBasicStringAppend() {
    String s = "";
    for(int i = 0; i < 100; i++) {
      s = new StringBuffer().append(s).append(i).toString();
    }
  }
}
```

It does not actually matter which compiler is better, either is terrible. The answer is to avoid += with Strings wherever possible.

Throw the used StringBuffers away!

You should never reuse a StringBuffer object. Construct it, fill it, convert it to a String, and then throw it away.

Why is this? StringBuffer contains a **char**[] which holds the characters to be used for the String. When you call toString() on the StringBuffer, does it make a copy of the **char**[]? No, it assumes that you will throw the StringBuffer away and constructs a String with a pointer to the *same **char**[] that is contained inside StringBuffer!* If you do change the StringBuffer after creating a String, it makes a copy of the **char**[] and uses that internally. Do yourself a favour and read the source code of StringBuffer - it is enlightning.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

But it gets worse than this. In JDK 1.4.1, Sun changed the way that setLength() works. Before 1.4.1, it was safe to do the following:

```
... // StringBuffer sb defined somewhere else
sb.append(...);
sb.append(...);
sb.append(...);
String s = sb.toString();
sb.setLength(0);
```

The code of setLength pre-1.4.1 used to contain the following snippet of code:

```
if (count < newLength) {
  // *snip*
} else {
  count = newLength;
  if (shared) {
    if (newLength > 0) {
      copy();
    } else {
      // If newLength is zero, assume the StringBuffer is being
      // stripped for reuse; Make new buffer of default size
      value = new char[16];
      shared = false;
    }
  }
}
```

It was replaced in the 1.4.1 version with:

```
if (count < newLength) {
  // *snip*
} else {
  count = newLength;
  if (shared) copy();
}
```

Therefore, if you reuse a StringBuffer in JDK 1.4.1, and any one of the Strings created with that StringBuffer is big, all future Strings will have the same size char[]. This is not very kind of Sun, since it causes bugs in many libraries. However, my argument is that you should not have reused StringBuffers *anyway*, since you will have less overhead simply creating a new one than setting the size to zero again.

This memory leak was pointed out to me by Andrew Shearman during one of my courses, thank you very much! For more information, you can visit Sun's website.

When you read those posts, it becomes apparent that JDOM reuses StringBuffers extensively. It was probably a bit mean to change StringBuffer's setLength() method, although I think that it is not a bug. It is simply highlighting bugs in many libraries.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

For those of you that use JDOM, I hope that JDOM will be fixed soon to cater for this change in the JDK. For the rest of us, let us remember to throw away used StringBuffers.

I hope to see some of you on the course next month where we look at some Java Design Patterns.

So long...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 069] - Treating Types Equally - or - Life's Not Fair!

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 69th edition of **The Java(tm) Specialists' Newsletter**. After my last newsletter, the listserver told me that quite a few of you did not receive the newsletter. If you are reading this on the internet, and you did not receive this newsletter, be warned - you have probably been deleted from the list! We now have approximately 5593 **working** email addresses.

Why can we not treat everyone equally? A few weeks ago, my four year old son Maximilian wanted to know the answer to that age-old question. Maxi's 3 year old cousin was visiting and was allowed to stay up and watch cartoons on TV, whilst he had to go to bed to sleep. "Sorry, Maxi, but as you will soon find out: **Life is not fair!** There is nothing I can do to change that, that is just the way it is."

A well known person (so well known that I have forgotten who it was) once said: "Everyone that I meet is my superior in some way"

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

This newsletter is about some operations who treat types equally (when perhaps they should not), and others, who due to their own feelings of inadequacy, do not. It all gets rather confusing, as you will soon find out. Remember, if you don't understand this newsletter: **Life is not fair!** Read the Java VM Spec. Write to the editor of this fine newsletter (editor@dev/null). But don't moan.

I want to thank all those who make the effort to write to me to correct errors in my newsletters. Please continue your good work. It makes the newsletter more useful when we remove the Gremlins.

For those living in Cape Town, South Africa, we are doing another Design Patterns Course in May 2003, please see the advert at the bottom of this newsletter. If you come on that course, you will also receive a free copy of the Java Specialists' Newsletters Book.

Treating Types Equally - or - Life's Not Fair!

What is the purpose of a **byte**?

Let me ask that question a bit differently: Why would you use a **byte**, as opposed to an **int**? To save memory? Only since JDK 1.4.x does a byte field take less memory than an int field. To increase computational speed? Maybe the opcodes for adding bytes are faster than for ints?

Perhaps **byte** should have been left out of the Java Programming Language? Not just **byte**, but also **short** and **char**? The original Oak specification, on which Java is based, had provision for unsigned integral values, but they did not make it, so why do we have those other types?

Ahh, but a **byte**[] will take up less space than a **char**[]. That is true, so they do have a reason to exist! But what about computation?

"Most of the instructions in the Java virtual machine instruction set encode type information about the operations they perform. For instance, the *iload* instruction loads the contents of a local variable, which must be an **int**, onto the operand stack. The *fload* instruction does the same with a **float** value. The two instructions may have identical implementations, but have distinct opcodes." - VM Spec The Structure of the Java Virtual Machine.

This brings along a slight problem: there are too many types in Java for the instruction set. Java has only one byte per opcode, so there are a maximum of 256 opcodes. There is therefore great pressure to have as few opcodes as possible.

The 18 opcodes that are defined for **int**, and not for **short**, **char** and **byte**, are: iconst, iload, istore, iinc, iadd, isub, imul, idiv, irem, ineg, ishl, ishr, iushr, iand, ior, ixor, if_icmpOP, ireturn. If these were also defined for the other three primitive types, we would require at least an additional 54 opcodes.

There is only one opcode that is marked as "unused". It is opcode 0xBA. Go figure. Probably the BSc Computer Science nerds having a dig at all the BA's that are unused ;-) *Fries with that?*

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

---

What does that mean for you and me? Let's look at a code snippet, sent to me by Jeremy Meyer. Jeremy helped me get my first job at a company called DataFusion Systems in South Africa, now called DataVoice, and part of a bigger company called Spescom. DataVoice are probably not hiring anyone at the moment, but they are one fine company to work for, so if ever you are offered a job there, take it at any price! Even working there for free would be a bargain, considering what you will learn there. Thanks Jeremy!

```
public class ByteFoolish {
  public static void main(String[] args) {
    int i = 128;
    byte b = 0;
    b |= i;
    System.out.println("Byte is " + b);
    i = 0;
    i |= b;
    System.out.println("Int is " +i);
  }
}
```

When we run this, we see the following:

```
Byte is -128
Int is -128
```

Here we start with a value bigger than 127 (the maximum positive **byte** value). We store it in an **int**, and then OR the bits into the **byte**.

The first System.out.println statement naturally reports the value of the **byte** as -128, which one would expect. The **byte** is, after all, signed and has range -128 to +127. Then we reset the **int** to 0, and OR the value of the **byte** back into the **int**.

The **int** now has the value of -128!

How could we OR the int with just the bits that belong to the last byte? We could first bitwise AND it with a mask that only shows the last byte and OR the result with the int.

```
public class ByteFoolish2 {
  public static void main(String[] args) {
    int i = 128;
    byte b = 0;
    b |= i;
    System.out.println("Byte is " + b);
    i = 0;
    i |= (b & 0x000000FF);
    System.out.println("Int is " +i);
  }
}
```

Now when we run the program, we see:

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
Byte is -128
Int is 128
```

Bitwise arithmetic with byte, short and char is challenging. Inside the JVM, these are first translated to ints, worked on, and then converted back to bytes. Let's disassemble the class to make sure that this is what is happening:

```
public class ByteFoolish3 {
  public ByteFoolish3() {
    int i = 128;
    byte b = 0;
    b |= i;
    i = 0;
    i |= b;
  }
}
```

We disassemble with javap -c ByteFoolish3 (you know how to do that by now):

```
 0 aload_0
 1 invokespecial #9 <Method java.lang.Object()>
 4 sipush 128   // push 128 onto stack
 7 istore_1     // store in int register 1
 8 iconst_0     // push constant "0" onto stack
 9 istore_2     // store in int register 2
10 iload_2      // load register 2
11 iload_1      // load register 1
12 ior          // OR them together as ints
13 i2b          // convert the int on the stack to a byte
14 istore_2     // store the value in register 2
15 iconst_0     // push constant "0" onto stack
16 istore_1     // store this in int register 1
17 iload_1      // load register 1
18 iload_2      // load register 2
19 ior          // OR them together
20 istore_1     // store them in register 1
21 return
```

Is i += n the same as i = i + n ?

I was sitting in a seminar on Refactoring by Martin Fowler a few years ago. The things Martin was saying sounded like music to my ears. I had refactored my code for many years, but had never heard such a thorough approach on the subject. The one thing that stuck in my mind was the difference between i += n and i = i + n.

Would the following compile?

```
public class Test1 {
  public static void main(String[] args) {
    int i = 128;
    double d = 3.3234123;
    i = i + d;
    System.out.println("i is " + i);
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

The answer is that it would not compile. A double is 64 bits with a very big range. There is no way that it would fit into an int without losing precision. It is not safe to run such code, so when we attempt to compile it, we get the following message:

```
Test1.java:5: possible loss of precision
found   : double
required: int
   i = i + d;
       ^
1 error
```

This is good. We pick up errors before we run the program. Ya'll know how to cast a double to an int if you definitely want to do that. You can either cast the values individually before you add, or you can cast the result:

```
public class Test2 {
  public static void main(String[] args) {
    int i = 128;
    double d = Integer.MAX_VALUE + 12345.33;
    i = i + (int)d;
    System.out.println("i1 is " + i);
    i = 128;
    i = (int)(i + d);
    System.out.println("i2 is " + i);
  }
}
```

In a way, I would expect both i's to have the same value, but due to the precision loss of doubles, they are not equal:

```
i1 is -2147483521
i2 is 2147483647
```

Let's have a look at the next class, Test3:

```
public class Test3 {
  public static void main(String[] args) {
    int i = 128;
    double d = Integer.MAX_VALUE + 12345.33;
    i += d; // oops, forgot to cast!
    System.out.println("i is " + i);
  }
}
```

Does this compile? Ooops, we forgot to cast! But, does it compile? Yes it does, and when we run it, we get:

```
i is 2147483647
```

Therefore, we can say that i += n is the same as i = (type_of_i)(i + n).

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Kind regards, and thanks for the great feedback after the last newsletter!

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 069b] - Results of last survey

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

This is a quick follow-up to let you know the score of the survey from the last newsletter. In the newsletter I mentioned that i += n is the same as i = (type_of_i)(i + n). A number of readers were surprised that I should mention such an obvious bit of information in an advanced Java newsletter. They were in the minority. Only 32 of the responses indicated that they already knew that information, and a staggering 174 readers responded that they did not know. That comes to only 15.53% of **Java Specialists** knowing about this. Imagine how few Java beginners know about this?

The quote was attributed to Ralph Waldo Emerson.

Beno Patrik from CLEVERLANCE wrote, suggesting that we could add the cast from **int** to **double** to the formula. That would mean that i += n is the same as i = (type_of_i)((type_of_n)i + n), if n is of higher precision than i. I did realise that when I was writing the newsletter, but did not want to muddy the waters. That is a typical upcast that we would expect. The one that I mentioned was a downcast, losing precision, and that we would not expect without a compiler error.

I have found a great free site for preparing online for the SCJP examination.

Yesterday, Andrew Righthouse and I spent the public holiday updating the "Feeding the 5000" webpage. We've added new pictures, and Andrew explained a bit more about what the people of the townships experience daily. South Africa is a concentrated example of the sufferings of this world. Less than 10km away from where I live is poverty where people live below the breadline. Visitors who come to

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

South Africa are disturbed about this; that some should live in big houses, drive fancy cars, and others in the same town would be starving. It is dreadful, a legacy of an evil system that will take years to right. Yet the contrast that is so evident in South Africa is a picture of the suffering of the whole world. We have rich countries and poor countries. They are perhaps separated by 1000's of kilometers, but that does not make the poor nations disappear. They are still there, whether you like it or not.

Lastly, but most importantly, I owe you a **big** apology. The advert for my Design Patterns Course was worded flippantly and insensitively. I apologise and have reworded it.

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 070] - Too many dimensions are bad for you

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 70th edition of **The Java(tm) Specialists' Newsletter**, where we have a quick look at multi-dimensional arrays. Many articles are available about this problem, so in this newsletter, I will briefly point out the problem and give you one example. I would like to thank Roy Emmerich (Peralex in Cape Town) and Pieter Potgieter and Louis Pool from Grintek Ewation for reminding me of this problem and for sending me a test program that illustrated it.

This past week we had a Design Patterns Course at one of the finest hotels of Cape Town. OK, it is *renowned* to be one of the finest ;-) The lunches were fantastic, but the coffees, well! Let me say that they are probably not used to having programmers at their hotel? There is a direct correlation between quality of coffee and quality of code, so we are quite discerning when it comes to the magic brew. But the lunches more than made up for it! On the first day we had baked sirloin steak, on the second we had filled chicken breasts and on the third we were treated to

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

excellently prepared line fish (Cape Salmon, otherwise known as Geelbek). The discussions about the Design Patterns were deep, and we all left thoroughly exhausted (especially me!).

We now have a schedule of when we would like to do Design Patterns and Java Courses in various places on planet earth (no plans for Mars yet). If your country is not on the list, please persuade us to add it :-) The schedule is on http://www.javaspecialists.co.za/courses/schedule.html.

Too many dimensions are bad for you

Something you learn in Java nursery school is to avoid multi-dimensional arrays like the plague. An array is an object in its own right, and when you have multi-dimensional arrays, you have arrays of objects. Navigating these will take considerable processing power. Instead of making a multi-dimensional array of size n by m, rather make a one-dimensional array of size n times m and then do the index calculation yourself.

```
public class MultiDimensions {
  private final static int NUM_BINS = 1000;
  private final static int ITERATIONS = 10000;

  public static void main(String[] args) {
    testMultiArray();
    testMultiArray2();
    testSingleArray();
  }

  private static void testMultiArray() {
    long time = -System.currentTimeMillis();
    // just making sure that the number of operations is equal
    int ops = 0;

    for (int repeat = 0; repeat < ITERATIONS; repeat++) {
      int[][] aTwoDim = new int[NUM_BINS][4];
      for (int i = 0; i < aTwoDim.length; i++) {
        for (int j = 0; j < aTwoDim[i].length; j++) {
          ops++;
          aTwoDim[i][j] = j;
        }
      }
    }

    time += System.currentTimeMillis();
    System.out.println(ops);
    System.out.println("Time Elapsed for [][4] - " + time);
  }

  private static void testMultiArray2() {
    long time = -System.currentTimeMillis();
    int ops = 0;

    for (int repeat = 0; repeat < ITERATIONS; repeat++) {
      int[][] aTwoDim = new int[4][NUM_BINS];
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    for (int i = 0; i < aTwoDim.length; i++) {
      for (int j = 0; j < aTwoDim[i].length; j++) {
        ops++;
        aTwoDim[i][j] = j;
      }
    }
  }

  time += System.currentTimeMillis();
  System.out.println(ops);
  System.out.println("Time Elapsed for [4][] - " + time);
 }

 private static void testSingleArray() {
  long time = -System.currentTimeMillis();
  int ops = 0;

  for (int repeat = 0; repeat < ITERATIONS; repeat++) {
   int[] aOneDim = new int[NUM_BINS * 4];
   for (int i = 0; i < aOneDim.length/4; i++) {
     for (int j = 0; j < 4; j++) {
       ops++;
       aOneDim[i* 4 + j] = j;
     }
   }
  }

  time += System.currentTimeMillis();
  System.out.println(ops);
  System.out.println("Time Elapsed for [] - " + time);
 }
}
```

When I run this code with the client hotspot under JDK 1.4.1, I get the following results:

```
40000000
Time Elapsed for [][4] - 4226
40000000
Time Elapsed for [4][] - 631
40000000
Time Elapsed for [] - 671
```

The server hotspot fares slightly better:

```
40000000
Time Elapsed for [][4] - 3675
40000000
Time Elapsed for [4][] - 350
40000000
Time Elapsed for [] - 561
```

The results are enlightning. If you have an array with a big first index, you will be better off using a single-dimensional array. Under server hotspot, the multi-

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

dimensional array with a small first index fares quite a bit better than the single-dimensional array.

That's all that I would like to say about this issue. I warned you that this newsletter would be short :-)

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 070b] - Multi-Dimensional Arrays - Creation Performance

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

One of our subscribers, Martin Schulte from Germany, sent me a correction for yesterday's newsletter, and I do not want to take too long before publishing it, since it is rather significant.

Martin discovered that the biggest difference in performance is the **creation** of the multi-dimensional arrays. Mark Grand also thought I should point out that there are no multi-dimensional arrays in Java, only arrays of arrays. This is the reason why we have a difference in performance when we create these arrays of arrays.

The difference in creation times can be seen in this small test example, I have left out the syntax highlighting to see your reaction ;-)

```
public class CreationTimes {
  private final static int ITERATIONS = 1000000;
  private final static int NUM_BINS = 20;
  private final static int THE_OTHER_DIM = 4;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
 public static void main(String[] args) {
   testMultiArray();
   testMultiArray2();
   testSingleArray();
 }

 private static void testMultiArray() {
   long time = -System.currentTimeMillis();

   for (int repeat = 0; repeat < ITERATIONS; repeat++) {
     int[][] aTwoDim = new int[NUM_BINS][THE_OTHER_DIM];
   }

   time += System.currentTimeMillis();
   System.out.println("Time Elapsed for [][" + THE_OTHER_DIM + "] - " +
                time);
 }

 private static void testMultiArray2() {
   long time = -System.currentTimeMillis();

   for (int repeat = 0; repeat < ITERATIONS; repeat++) {
     int[][] aTwoDim = new int[THE_OTHER_DIM][NUM_BINS];
   }

   time += System.currentTimeMillis();
   System.out.println("Time Elapsed for [" + THE_OTHER_DIM + "][] - " +
                time);
 }

 private static void testSingleArray() {
   long time = -System.currentTimeMillis();

   for (int repeat = 0; repeat < ITERATIONS; repeat++) {
     int[] aOneDim = new int[NUM_BINS * THE_OTHER_DIM];
   }

   time += System.currentTimeMillis();
   System.out.println("Time Elapsed for [] - " + time);
 }
}
```

The result of running this code is:

```
Time Elapsed for [][4] - 9653
Time Elapsed for [4][] - 2394
Time Elapsed for [] - 671
```

On another note, I will be joining an elite group of authors on Bill Venners' Weblogs. The others on the list are really quite famous, so I am there to bring some balance :-) Check out the artima website on http://www.artima.com, I should be there within a week or so. I plan to use that forum as a supplement to my newsletter, and it will give you an opportunity to chat back publicly as well.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 071] - Overloading considered Harmful

Author: Alexander (Sascha) H˜her

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 71st edition of **The Java(tm) Specialists' Newsletter**. We have a new country on our subscription list; please welcome *the* **Faroe Islands** (applause)! I had never heard of them, but then again, my knowledge of Geography has never been good. There is a nice website about these islands, you can have a look at www.faroeislands.com. We are approaching 100 countries in our subscription list, and you can see whether your country is already in our count by looking on our web page under countries.

A few weeks ago, Alexander (Sascha) Hoeher sent me a piece he had written in German on method overloading. The writing was persuasive and interesting, so I asked Sascha if he could perhaps translate it into English so we could feature his ideas on this newsletter. The ideas were based on something that I had written about 2.5 years ago, on Depth-first Polymorphism, but Sascha added a dimension to the argument that I had missed. He asks whether perhaps overloading is harmful? I enjoyed both the German and the English version - I trust you will also. The format is a bit different to my usual newsletters, but I encourage you to spend the time reading through the prose and let use know what you think of this.

Alexander Hoeher is based in Weilheim near Munich, Germany. After getting into software development with C++, he found himself attracted to Java, seeing its simplicity as a necessary basis for architecting reliable complex systems. A Sun Certified Enterprise Architect by now, his ideas about object-oriented design have been strongly influenced by playing around with Eiffel and digging into UML literature (the latter resulting in an IBM UML certificate). His current interests comprise

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

concurrent programming (formal process modeling), design patterns, and XML related topics. He is fond of juggling, monocycling, swimming, and a dedicated guitar-player.

Overloading considered harmful

What is overloading, once again? Same method name for different methods - sounds harmless enough!

Sure it's one of the first things Java programmers are confronted with when learning the language. You are told things like: Do not mix it up with overriding - remember, these things may look quite similar, but are altogether different concepts! Then your Java introduction goes on telling you about unique parameter lists, and after one and half pages you get the impression that this is something not so terribly hard to understand. *[HK: I can vouch for this argument. In my Java courses, students commonly make this mistake.]*

What is the value proposition of this seemingly simple feature?

Shorter interfaces, not bogged down by artificial, tiresome discriminators, and a bit of lexical structuring of your class text: Overloading allows you to indicate the conceptual identity of different methods, letting you stress common semantics across methods so that similarities become apparent at first sight. It's supposed to make your code more readable, and what regards server code - the code, where these method siblings are defined -, it really does.

There are many who like it. There is tons of code using what overloading has to offer. And of course, you cannot even escape it in Java, where you're simply forced to use it when you want to provide different initializers. It seems, overloading rules - a feature not only popular, but tightly integrated into some important programming languages, an open commitment of venereous language designers that surely does not fail to impress the masses. And, what is more: no performance penalty whatsoever...

Now, should we fully embrace overloading in our own code then? Should we use it wherever possible? This discussion shall present an attempt to put the technical facts investigated in-depth by a former edition of this newsletter into a usage perspective - a bit similar in spirit to the popular harping on pointers which you can find in every Java introduction. The seminal idea that overloading clashes with dynamic binding is taken from a discussion of overloading to be found in "Object-Oriented Software Construction" by Bertrand Meyer.

There is no reason to question that naming conventions to indicate conceptual interrelatedness of different methods will benefit the class text where these methods are defined. To adopt the convention of reusing the same method name, however, has unfortunate consequences on client code which can become quite unintuitive, to say the least.

Overloading with parameter lists of different length pose no problem for client code interpretation, as they openly disambiguate client calls at first sight. Things that could irritate you just will not compile. However, when overloaded methods with the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

same method name have parameter lists of the same length, and when the actual call arguments conform to more than one signature of these overloaded methods, it somehow gets a little hard to tell which methods are actually executed just looking on the client calls. In this situation, you experience the strange phenomenon that the methods being called are not independent of the reference types being used for the calls.

There are several problems related to this, but first let's take another look on the small code example presented in a former edition of this newsletter in order to really get a feel for what it's like when methods being called are not independent of the reference types being used for the calls.

A minimal modification allows us to focus on the ugly side of overloading: The program still tells us which method gets actually called, but on top of that also delivers rather strong comments when overloading is caught to harm our ability to reason about the client code without knowing the server classes.

Basically, we have two fixed instances, which will play always the same roles: one serving as call target, the other serving as argument. Now we mix and match several calls always to be executed on these same instances (always the same target object, always the same argument object) the only difference being the reference types through which these objects are accessed. And behold: Different methods are being called. If you are familiar with this simple setting, you may skip the program part to directly go on with the following discussion.

```
public class OverloadingTest {
  public abstract static class Top {
    public String f(Object o) {
      String whoAmI = "Top.f(Object)";
      System.out.println(whoAmI);
      return whoAmI;
    }
  }

  public static class Sub extends Top {
    public String f(String s) {
      String whoAmI = "Middle.f(String)";
      System.out.println(whoAmI);
      return whoAmI;
    }
  }

  public static void main(String[] args) {
    Sub sub = new Sub();
    Top top = sub;

    String stringAsString = "someString";
    Object stringAsObject = stringAsString;

    if (top.f(stringAsObject) == sub.f(stringAsString))
    //if (top.f(stringAsObject) == sub.f(stringAsObject))
    //if (top.f(stringAsString) == sub.f(stringAsString))
    //if (top.f(stringAsString) == sub.f(stringAsObject))
    //if (sub.f(stringAsString) == sub.f(stringAsObject))
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    //if (top.f(stringAsString) == top.f(stringAsObject))
    {
      System.out.println("Hey, life is great!");
    } else {
      System.out.println("Oh no!");
    }
  }
}
```

Can you tell what happens with activating each of the conditions?

Let us carefully go through the code.

There are two overloaded methods spread across a class hierarchy (one class inheriting from another class). This is the server code to be called by the client.
The superclass defines: String f(Object o).
The subclass defines: String f(String o).
The signatures are chosen to make both methods eligible candidates to be executed in the context of calls on the subclass instance with a String argument.
The client provides two objects, reused for all calls and chosen in a way that both overloaded methods are potentially eligible candidates for executing the client calls. Through polymorphic assignment, the client obtains references of different types for these two instances.
The client makes method calls that differ only in the different references used for making the call. In the given setup, there are 4 different call forms possible: Overloading has the method name fixed, so only the target reference type and the parameter reference type are variable. Every reference type for the target can be combined with every reference type for the argument. (Mathematically spoken, there are 4 binary strings of length 2).
The comparisons then are really just for fun, eliminating detail. They shift the focus of attention from the question what particular method gets called to the general insight that different methods get called, additionally allowing the program to be explicit about its likes and dislikes: Every case of seeming reference-independence of the calls is instantly interpreted as an example of how things should be, and welcomed with a happy, optimistic "Hey, life seems great!" In those some dark moments, however, when overloading casts its dark shadow upon the else so object-oriented Java world, and just nothing seems right, our little program starts to complain... Combinatorics tells us six 2-combinations of a 4-set (consisting of 4 call forms) exist, and so you find six comparisons (five of them showing up as comments), but of course, already one single predicate returning false (different methods having been called) suffices to get the point across.

And that's it.

Discussion

The program shows, once again, that one thing to be aware of in connection with overloading is that it's all about reference types. This is as true for target reference types as it is for parameter reference types. For instance, the predicate "sub.f(stringAsObject) == sub.f(stringAsString)" will resolve to false in our setup because two different methods are executed. This dependence on reference types in connection with overloading may or may not be what you expected, but the question remains if this is a clean approach to object-oriented programming.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

No doubt, this may puzzle many a brave programmer, as it is a result absolutely exclusive to overloaded methods. And, as the use of overloaded methods does not identify itself as such in the method call, the intuitive, but unfortunately wrong expectation might be that the predicate returns true, as it would be the case with any gentle non-overloaded method.

Honest, do we like this? No. Object-oriented programming, as we know it, is about objects, not about references. We expect objects to behave the way they are and not the way they are referenced. Objects do their thing regardless of the role the client assigns them. This is how it should be be, and we call this thing dynamic binding. It is not cosmetics, it is not just a feature, it is THE feature. It shapes the architecture of our systems, decoupling clients from servers.

Now, with overloading a second rule, reference type dependence, takes over, breaking the fundamental polymorphic equivalence property described above (that polymorphic assignments do not change the results of method calls as long as the code can be compiled). The choice of references in the client, which should be based on considerations like grouping and low coupling, suddenly has to take the demands of overloading into account. Overloaded server objects affect the design of client code. Cosmetics beat structure. Unlike overriding, overloading cannot just be applied in a server method definition act and end of the story. It is a feature you have to stay aware of in your clients whose specific referencing of server objects influences what functionality gets called in the end. While with dynamic binding alone the method to be executed is completely server-defined, overloading proves to be client-sensitive.

Now to the problems. An important issue closely connected to software quality is readability. Our ability to reason about the software text is essential for any kind of maintenance, and, as you might have guessed by the direction this discussion has taken by now, overloading affects readability of client code in a rather negative way. It is all very well to let the program run and after the surprise look at the server code and explain the strange things away (oh, of course, overloaded methods, you know...), but nevertheless it would be preferable by far to predict the behaviour, simple as it is, by simply (i.e. exclusively) examining the client. Show me the client class, tell me no overloading is involved, and I tell you: "Hey, life seems great!" I can reason about the result of the condition solely looking at the client class.

With overloading being introduced, or even with just the slightest chance of overloading being used (this includes all unknown Java code), this statement is impossible to make, because you cannot tell if the same server method gets called without examining the server sources. In our program, you would have to read three classes instead of one to know what's going on. So, use of overloading weakens the expressive power of client code as the polymorphic equivalence property cannot be relied upon.

Sometimes, of course, you are willing to dig into the server code because you want to find out the exact server method that gets called. But even then overloading significantly complicates things. Without overloading, you just work your way bottom up through the target's class hierarchy, and when you find a match, bingo, you're done! With unknown code or code known to use overloading, this can be only your second step. First you have to examine the class of the reference and find the matching method. Only then can you check the class hierarchy for overriding methods. The bad thing about this is probably not the additional step involved, but that you have to repeat this analysis for every different reference type, because

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

results can vary. Thus, overloading complicates the analysis of client-server interaction.

There is also a psychological dimension to all this. The following will try to show that overloading is not a gentle, unobtrusive language feature, but, as it stands in conflict with other language features, late binding and inheritance, particularly prone to abuse. In other words, overloading is an open invitation for introducing conceptual errors. Think of novice programmers or programmers in a rush. Overloaded methods, coming with its own method selection rules, present an anomaly in the object-oriented landscape shaped by the presence of dynamic binding, and will surely go on to puzzle people, who will falsely think overloaded methods behave like "normal" methods, or mistake overloading for overriding just because the methods signatures involved in overloading look so similar.

In fact, such a mistake may be seen as expressing justified desires regarding object-oriented design. Hell, we'd sure like to see the overloaded methods in our example being handled as an instance of overriding! The parameters of our methods are related through inheritance, so inspired by other programming languages, it does not take great imagination to see the derived class define a method that overrides the inherited method. Of course, this is an additional twist adding a bit of vision to our discussion, and of course, we know that Java does not support such covariant method redefinition (restricting the parameter domain of the method): Most of us have learnt by now that Java allows only specification inheritance (overriding being only defined for methods with the same return and parameter types). But still. Do we not think, deep in our heart, that the subclass method with the more specific parameter should, in a better world maybe, be the one in charge, overriding the superclass method? Think about an Integer class inheriting from Number while redefining addition for integers only. Not allowed in Java, but still desirable (and a real feature in other languages such as Eiffel). Sure, overloading is not to be blamed for an incorrect understanding of inheritance in Java, but it clearly invites such fantasies (and the corresponding errors) when used in a context such as the presented one. And even if such interpretations are wrong - shouldn't they be right?

And then the poor integration of overloading and inheritance in Java, which is very misleading as well. Reference type dependence means that overloading is simply not developed to conceptual consistency in the context of inheritance. Guessing from experience with overloaded methods defined in one and the same class, we might expect the method with the best match in terms of formal parameter type and actual method argument to be called on the object. This does not happen, though. Java does not produce any kind of "flat form" for the object's class with all overloaded methods, inherited or not, appearing side by side in a list in order to allow the runtime to choose the most appropriate.

No, what technically happens, is, in my understanding, that the compiler takes the method symbol plus the parameter reference types of some method call and calculates a position in the method table of the target's reference type. So, choosing between overloaded methods is done compile-time, and it is restricted to the overloaded methods of one class: the class of the reference type. Overloaded methods defined in subclasses of the reference type are never called: Java ignores the exiled siblings although the whole thing looks so very similar to overriding.

With overloaded methods being defined in superclasses of the reference type, Java exhibits quite strange behaviour: While the server code can still be compiled, client code will break: Trying to make a method call where the compiler would have to

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

choose between them, you get a compilation error, complaining that the call is ambiguous. Put the method into the reference type and all is well. Don't ask me why - just remember selection of overloaded methods is limited to the reference type class. I personally believe this further anomaly might is more a compiler issue than a language issue. If you find a logic explanation for this, other than that it helps to improve compilation performance, please let me know.

Once again (the last time): overloaded methods defined in subtypes of the target reference will not be taken into consideration as candidates for execution by the runtime. With the table position given in the bytecode, the runtime will only check if there are overriding methods (which will appear at the same position in the method tables of subclasses if they exist). So, the compiler cannot hunt them down, and the runtime does not want to.

A consequence of this is, disturbingly, that the place where non-overridden overloaded methods are defined in the class hierarchy is of essential importance what regards the selection of the method being actually called. To me, this sounds a little scary, or would you really want your class design to be influenced by the crippled demands of overloading? Summing up: Overloading is a static compile-time feature which does not integrate well with our expectations shaped by dynamic method lookup coming along with inheritance.

What else can we do to shoot the dead man? (Who is still alive enough to ruin our programs, of course.) Bertrand Meyer sees overloading as a violation of the "principle of non-deception: differences in semantics should be reflected by differences in the text of the software" (OOSC 94, Bertrand Meyer). But wait a second, isn't late binding another case where there is only one method symbol for different methods?

As I understand it, the difference between late binding and overloading can be pinned to the observation that late binding lets one method name to be the pointer to one operation contract (which then can be fulfilled by several different methods whose differences are nevertheless absolutely transparent to the client code), whereas overloading lets one method name to be the pointer for several method specifications whose differences can be experienced in the client code. In the scope of the client, there is no difference between polymorphic calls bound to different methods. The polymorphic call specification is all the client has to know about the call. Overloaded methods, on the other hand, need not share common semantics, to be more precise, a common contract, their pre- and postconditions potentially varying wildly. This is something the client always has to take into account: Overloaded methods can not be used interchangeably, as different methods just under the same hood they have to be treated according to their specific contracts. These contracts, however, are hidden behind the same name which makes them hard to identify.The same method name does not point to a common denominator, in this case, but only serves to disguise differences that have to be laboriously disambiguated lateron. The client has to stay aware of the method contract being pointed to by a complicated three component key for the method which, as we have seen, consists of target reference type, method name, and parameter reference types.

So what are my final words to the programmer who, after having read this article, wonders if he should try to use overloading now wherever possible or not? Keep going... And if you really, really want to use it, go on and do so, but only with

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

different method names - this is a trick stolen from real experts that can improve your overloading a lot! :o)

Sascha

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 072] - Java and Dilbert

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 72nd edition of **The Java(tm) Specialists' Newsletter**. This time we welcome a subscriber from Uganda, bringing our total countries to 97. Three more to go and we will have reached 100 countries. Perhaps we will even reach 100 countries by the end of this year :-)

Switzerland, July 2003: During the first two weeks of July, I will be in Switzerland on business, and it would be great to meet up with some fellow Java freaks. Please pop me an email so we can try and coordinate something.

Durban, 19th June 2003: I will be in Durban, South Africa, at the newly formed Java User Group talking about Design Patterns. Please look at http://www.dbnjug.org.za for more information.

Java and Dilbert

Programmers are obliged to read Dilbert. One says that a picture is worth a thousand words, and Scott Adams certainly captures our day-to-day frustrations in his comic. I once verbalised my enthusiasm for Dilbert to a boss, who then suggested that I was like one of the characters, but not necessarily Dilbert. By process of elimination, I have concluded that he was probably thinking of Wally, my hero :-)

Many years ago, I read a book called "Cheaper by the Dozen" by Frank Gilbreth, about a time-and-motion expert and his wife with their twelve kids. The father was

an early version of a process reengineering expert. In his study of companies, he would pick the laziest person to study, because they would have already optimised the process to suit their own laziness. For example, he should look at the person who will rather write a batch file once than to repeat the same command over and over again. Windows NT (and beyond) has an autocomplete option in the command prompt, so the person that has that enabled in the registry would be a good candidate.

Wally would be the person I would make in charge of automating the build process of the code, and running the unit tests. You can bet that there would be no manual intervention necessary at all.

So, what does all this have to do with Java? Java interfaces nicely with the internet, so any tasks that you would have to do repetitively over and over again can be automated quite easily. You can interface with any URL by simply passing it in as a String to the java.net.URL class and then calling openStream(). With a simple **while** loop, you can read the bytes from the URL and write them to a local file. You can convert this URL to an image with the javax.swing.ImageIcon. If the URL is pointing to an HTML file, you can search for certain strings, such as mailto:. [That last suggestion is used by spammers. Amazing how many African statespeople have vast amounts of money lying around at the moment that they want to dispose of. I get at least one of those spams a week. Talking of Spamming, I get challenge-response emails from some subscribers which they use to filter spams. Please add my email address to your acceptable filter if you want to receive my newsletter.]

Let us get back to Dilbert. Imagine Wally wanted to read Dilbert every day. Do you really think that Wally would connect every day to the internet and go to http://www.dilbert.com? That would be far too much effort! Instead, he would write a small Java program that went to the main Dilbert page, parsed the HTML page to find out what today's GIF URL is, and show it in an ImageIcon. An added benefit is that Wally is then not exposed to the advertising anymore. Here's the code:

```
import java.io.*;
import java.net.URL;
import javax.swing.*;

public class DilbertComic {
  public static String todaysDilbert() throws IOException {
    // open up the webpage to today's comic
    URL url = new URL("http://www.dilbert.com");
    BufferedReader webRead = new BufferedReader(
      new InputStreamReader(url.openStream()));
    String line;
    while ((line = webRead.readLine()) != null) {
      if (line.indexOf("ALT=\"Today's Dilbert Comic\"") != -1) {
        int offset = line.indexOf(
          "<IMG SRC=\"/comics/dilbert/archive/images/dilbert");
        line = line.substring(offset + 10);
        return "http://www.dilbert.com" +
            line.substring(0, line.indexOf("\""));
      }
    }
    return null;
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
 /**
  * This would allow us to download the URL to a local file.
  * It is so easy that we do not need any explanation :-)
  */
 public static void download(URL url, File file) throws IOException {
  InputStream in = url.openStream();
  FileOutputStream out = new FileOutputStream(file);
  byte[] b = new byte[1024];
  int len;
  while((len = in.read(b)) != -1) {
   out.write(b, 0, len);
  }
  out.close();
 }
 public static void main(String[] args) throws IOException {
  System.out.println("Looking for today's dilbert comic . . .");
  String today = todaysDilbert();

  if (today == null) {
   System.out.println("Could not find today's dilbert comic!");
  } else {
   System.out.println("Found today's dilbert: " + today);
   URL url = new URL(today);

   // we could download the comic to a local file like this:
   // download(url, new File("todaydilbert.gif"));

   // Instead, we are simply going to download it as an ImageIcon
   // and show it in a JFrame.

   System.out.println("Downloading the Image . . .");
   ImageIcon im = new ImageIcon(url);
   System.out.println("Downloaded the Image");

   JFrame f = new JFrame("Today's Dilbert");
   f.getContentPane().add(new JLabel(im));
   f.pack();
   f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   f.show();
  }
 }
}
```

The program could be upgraded to do this automatically once a day. We could save the image for future viewing (since the Dilbert webpage only allows you to see one month's worth of archives).

We did not cover anything earth-shattering in this newsletter, I know. However, when I have shown this to Java programmers, they have, for some reason, been surprised how easy it is to do something like this. I have used this technique to autodownload Garfield going back to 1978, Zapiro (Warning, Zapiro's cartoons has been deemed **offensive** by some of the most powerful countries in the world. Only read it if you are not sensitive or patriotic - you've been warned!) and a South

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

African cartoon called Madam and Eve that deals with the tensions in South African households.

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 073] - LinkedHashMap is Actually Quite Useful

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 73rd edition of **The Java(tm) Specialists' Newsletter**. Five years ago today, I was holding my newly-released son Maximilian Francis in my arms.

After last week's newsletter, some readers remarked that Dilbert could also be received via email. I knew that, but my way is more interesting than subscribing to a mailing list and has far greater applications than just "Dilbert". Another reader suggested that we should always read the "terms of use" of websites to ensure we do not inadvertantly violate a law.

Switzerland, July 2003: During the first two weeks of July, I will be in Switzerland on business, and it would be great to meet up with some fellow Java freaks some place near Zurich for a social evening. Please pop me an email so we can try and coordinate something.

LinkedHashMap is Actually Quite Useful

Did you know that the JDK 1.4.x contained a LinkedHashMap and a LinkedHashSet?

I love catching people off-guard. I love it more than being caught off-guard. I was speaking to my pastor the other day and told him: "Bruce, you should put a wireless network in your house so that you can communicate with your office downstairs."

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Bruce had this look on his face of "Yeah, pull the other leg, I won't fall for you again!" That is a problem that wisecracks experience: even useful comments sound like jokes :-(

This is similar to my experience with programmers when I mention the LinkedHashMap. It usually results in a look of puzzlement: "Should I believe that such a thing exists? Or is Heinz having me on again?"

Granted, the term LinkedHashMap sounds like something I would invent, like the IdentityWeakSoftHardPhantomHashLinkMap that I spoke about in an earlier newsletter.

What is a LinkedHashMap?

A LinkedHashMap is a combination of hash table and linked list. It has a predictable iteration order (a la linked list), yet the retrieval speed is that of a HashMap. The order of the iteration is determined by the insertion order, so you will get the key/values back in the order that they were added to this Map. You have to be a bit careful here, since re-inserting a key does not change the original order.

Let us imagine that we want to retrieve rows from a database table and convert these rows to Java Objects. Let us also imagine we are not using JDO. *[Incidentally, we have our own JDO implementation **Made in South Africa**, JDO Genie. A buddy from university is one of the directors, and he has agreed to do a review of the O'Reilly book on JDO for us soon :-]* We need the elements in some order, and we need to search on keys quickly.

Pre JDK 1.4, I would probably have made the objects Comparable and inserted them into a TreeMap. TreeMap works with a type of balanced binary tree, so the lookup would be O(log2n). If we have 1024 elements, it will take at most ten lookups. This is worse than a HashMap with a good hash function, which should only take one lookup. Keeping the tree balanced is expensive since it involves shuffling the nodes around whenever one side of the tree becomes too deep. Inserting the elements into a TreeMap is probably more expensive than sorting in your database, since the database is optimised for such things. [Balanced binary trees make excellent second-year computer science assignments. Mine did not work, but then neither did my tutor, so he gave me full marks.]

With the LinkedHashMap we get the best of both worlds. We can let the database do the sorting, retrieve the rows and insert them into the LinkedHashMap. This way we can retrieve them quickly via the HashMap mechanism, but we can still iterate through them in the sorted order.

To illustrate, let us use a domain that nerds know little about: Sport. We start with a Player, which we term as someone professionally engaged in a particular sporting activity. He makes his money from running around, instead of pushing a mouse around a mousepad. My father-in-law was a professional sportsman and even at sixty he is fitter than me. My brothers-in-law are world-ranked gravity racers. I was excused from physical education.

```
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
/**
 * Super athlete who earns money running around and styling
 * his hair nicely.
 */
public class Player {
  private final Key key;
  private final String name;
  private final Date dateOfBirth;
  public Player(String id, String name, Date dob) {
    this.key = new Key(id);
    this.name = name;
    this.dateOfBirth = dob;
  }
  public Key getKey() {
    return key;
  }
  public String getName() {
    return name;
  }
  public Date getDateOfBirth() {
    return dateOfBirth;
  }
  private final static DateFormat df =
    new SimpleDateFormat("yyyy/MM/dd");
  public String toString() {
    return name + " born on " + df.format(dateOfBirth);
  }
  public static final class Key {
    private final String id;
    public Key(String id) {
      if (id == null) {
        throw new IllegalArgumentException();
      }
      this.id = id;
    }
    public int hashCode() {
      return id.hashCode();
    }
    public boolean equals(Object obj) {
      if (!(obj instanceof Key)) return false;
      return id.equals(((Key) obj).id);
    }
  }
}
```

We now define a SportDatabase interface:

```
/**
 * This represents some database that retrieves the Players from
 * a backend database.
 */
public interface SportDatabase {
  Player[] getPlayers();
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

In South Africa, we put on white clothes and stand on a big field hoping for a ball to come our way. The complaining outfielders kept on "chirping", which led to the sport being called "Cricket".

```
import java.util.Date;
/**
 * Cricket is a boring sport that is popular in countries
 * previously occupied by the British Empire.  Like warm
 * beer, the taste for cricket has to be acquired.  It is
 * probably more of a spectator sport than unterwater hockey,
 * but not by much.
 *
 * South Africa does officially have a cricket team.
 */
public class CricketDatabase implements SportDatabase {
  private final static Player[] p =  {
    new Player("12341", "Boeta Dippenaar", new Date(77, 5, 14)),
    new Player("23432", "Gary Kirsten", new Date(67, 10, 23)),
    new Player("23411", "Graeme Smith", new Date(81, 1, 1)),
    new Player("55221", "Jonty Rhodes", new Date(69, 6, 27)),
    new Player("61234", "Monde Zondeki", new Date(82, 6, 25)),
    new Player("23415", "Paul Adams", new Date(77, 0, 20)),
  };
  public Player[] getPlayers() {
    return p;
  }
}
```

Note that the players are sorted by name. We assume that the players would be retrieved in that sort order from the database. We can see the difference between the HashMaps from this example:

```
import java.util.*;
public class LinkedHashMapTest {
  private static void fill(Map players, SportDatabase sd) {
    Player[] p = sd.getPlayers();
    for (int i = 0; i < p.length; i++) {
      players.put(p[i].getKey(), p[i]);
    }
  }
  private static void test(Map players, SportDatabase sd) {
    System.out.println("Testing " + players.getClass().getName());
    fill(players, sd);
    for (Iterator it = players.values().iterator(); it.hasNext();) {
      System.out.println(it.next());
    }
    System.out.println();
  }
  public static void main(String[] args) {
    SportDatabase sd =  new CricketDatabase();
    test(new HashMap(), sd);
    test(new LinkedHashMap(), sd);
    test(new IdentityHashMap(), sd);
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

When we run this code, we get the following output:

Testing java.util.HashMap
Jonty Rhodes born on 1969/07/27
Graeme Smith born on 1981/02/01
Paul Adams born on 1977/01/20
Monde Zondeki born on 1982/07/25
Gary Kirsten born on 1967/11/23
Boeta Dippenaar born on 1977/06/14

Testing java.util.LinkedHashMap
Boeta Dippenaar born on 1977/06/14
Gary Kirsten born on 1967/11/23
Graeme Smith born on 1981/02/01
Jonty Rhodes born on 1969/07/27
Monde Zondeki born on 1982/07/25
Paul Adams born on 1977/01/20

Testing java.util.IdentityHashMap
Paul Adams born on 1977/01/20
Jonty Rhodes born on 1969/07/27
Gary Kirsten born on 1967/11/23
Graeme Smith born on 1981/02/01
Monde Zondeki born on 1982/07/25
Boeta Dippenaar born on 1977/06/14

LRU Cache

Another application for the LinkedHashMap is in building LRU caches. One of the constructors allows us to specify that we use access order instead of insert order. The order is from least-recently accessed to most-recently. You can override the removeEldestEntry(Map.Entry) method to impose a policy for automatically removing stale when new mappings are added to the map. The implementation of this LRUCache is left as an exercise to the reader, or just look at the IdentityWeakSoftHardPhantomHashLinkMap from our earlier newsletter ;-)

Kind regards

Heinz

P.S. In case you thought I was lying about my brothers-in-law being world-ranked street lugers and skateboarders, search for Kytides on the gravity-sports world rankings. Sponsor them now, and you might become famous with them ;-)

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 074] - GoF Factory Method in writing GUIs

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 74th edition of **The Java(tm) Specialists' Newsletter**. I have had complaints by readers that their SPAM filters think my newsletters are junk mail. Thanks to one of my readers, I have eliminated two characteristics of my newsletter that seem to trigger these false alerts. Please let me know if your SPAM filter still warns you - and if possible also tell me *why it moans*.

Switzerland was great! We had some excellent discussions around Java, and I met some very smart people indeed. It forced me to spend some more time reading up on the latest trends, one of which is Java Data Objects (JDO). With the latest release of JBoss 4, we now have both JDO **and** a reliable multicast based peer-to-peer JMS implementation. In addition to JBoss starting to ship with JDO, Jakarta now also have a project that includes JDO. To me, JDO is one of the most significant developments in the Java standards in a long time, even though it will be an optional standard in J2EE.

GoF Factory Method in writing GUIs

Programmers often amazed me by claiming that Eclipse is not as good as XYZ IDE because it does not have a GUI editor. That seems to be the only weakness with Eclipse, so I am led to believe.

Seen from another angle, when I ask programmers which Design Patterns they know, I usually hear either **Singleton**, **Factory** or **Facade**. **Singleton** is usually an excuse to make a procedural design look more object-orientated. There is no pattern called **Factory** in the Gang-of-Four book. There is an **Abstract Factory** and a **Factory Method**, both of which are completely different to the *"static method that creates objects"* idiom. There is a lot of confusion about this, even in the Refactoring literature this is mixed up. **Facade** in the Gang-of-Four is in my estimation not really a Design Pattern, rather, it is an idiom that one can use when applying too many Design Patterns results in an overly complicated design. I have more arguments in my arsenal for these statements, but let us not argue. If you disagree, you can either join me on my Design Patterns Course, or we can just agree to disagree...

The problem with RAD GUI Tools

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Rabid Application Development. It promised to save millions of development dollars by replacing developers with wizards. However, in reality these wizards generate terrible, downright dangerous, code. We developers then spend more time improving it, than it would have taken, had we used our grey matter in the first place.

But that is not the worst problem with RAD GUI Tools. The worst problem is that they encourage the Copy & Paste Antipattern, where each time we develop a dialog, we start with a blank canvas. We do not think about reusing parts of the code, since the auto-generated code is difficult to work with. In addition, if we change too much of the code, we will not be able to work with the RAD GUI tool anymore. Usually all goes well until we have about 30 dialogs, and all of a sudden, the wheels come off this model. The code becomes too complicated to maintain, and eventually we move over to a web-based GUI so that we can start on a clean slate (and *without* RAD GUI tools!)

Let us look at an example, of some code that was written with evil wizards using an IDE that has been around for a long time. I have been a staunch supported of this unnamed IDE since version 1, and I can tell you that there have been almost no improvements in the GUI editor from version 3 to version 8 of the IDE. The IDE is great in general, but look at the code that is generated:

```
import javax.swing.UIManager;
import java.awt.*;

public class Application1 {
  boolean packFrame = false;

  //Construct the application
  public Application1() {
    Frame1 frame = new Frame1();
    //Validate frames that have preset sizes
    //Pack frames that have useful preferred size info, e.g. from their layout
    if (packFrame) {
      frame.pack();
    }
    else {
      frame.validate();
    }
    //Center the window
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    Dimension frameSize = frame.getSize();
    if (frameSize.height > screenSize.height) {
      frameSize.height = screenSize.height;
    }
    if (frameSize.width > screenSize.width) {
      frameSize.width = screenSize.width;
    }
    frame.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height -
frameSize.height) / 2);
    frame.setVisible(true);
  }
  //Main method
  public static void main(String[] args) {
    try {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
    }
    catch(Exception e) {
      e.printStackTrace();
    }
    new Application1();
  }
}
```

And the frame that contains all the components looks like this:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;

public class Frame1 extends JFrame {
  JPanel contentPane;
  BorderLayout borderLayout1 = new BorderLayout();
  JPanel jPanel1 = new JPanel();
  JLabel jLabel1 = new JLabel();
  JScrollPane jScrollPane1 = new JScrollPane();
  JPanel jPanel2 = new JPanel();
  JButton jButton1 = new JButton();
  JButton jButton2 = new JButton();
  JTable jTable1 = new JTable(new DefaultTableModel(100, 20) {
    public String getColumnName(int column) {
      return Integer.toString(column+1);
    }
    public Object getValueAt(int row, int column) {
      return Integer.toString((row+1)*(column+1));
    }
  });

  //Construct the frame
  public Frame1() {
    enableEvents(AWTEvent.WINDOW_EVENT_MASK);
    try {
      jbInit();
    }
    catch(Exception e) {
      e.printStackTrace();
    }
  }
  //Component initialization
  private void jbInit() throws Exception  {
    contentPane = (JPanel) this.getContentPane();
    contentPane.setLayout(borderLayout1);
    this.setSize(new Dimension(603, 483));
    this.setTitle("Frame Title");
    jLabel1.setText("Multiplication Table");
    jButton1.setText("OK");
    jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
      public void mouseEntered(MouseEvent e) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      jButton1_mouseEntered(e);
    }
    public void mouseExited(MouseEvent e) {
      jButton1_mouseExited(e);
    }
  });
  jButton2.setText("Help");
  jButton2.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(MouseEvent e) {
      jButton2_mouseClicked(e);
    }
  });
  contentPane.add(jPanel1, BorderLayout.NORTH);
  jPanel1.add(jLabel1, null);
  contentPane.add(jScrollPane1, BorderLayout.CENTER);
  jScrollPane1.getViewport().add(jTable1, null);
  contentPane.add(jPanel2, BorderLayout.SOUTH);
  jPanel2.add(jButton1, null);
  jPanel2.add(jButton2, null);
 }
 //Overridden so we can exit when window is closed
 protected void processWindowEvent(WindowEvent e) {
  super.processWindowEvent(e);
  if (e.getID() == WindowEvent.WINDOW_CLOSING) {
    System.exit(0);
  }
 }

 void jButton1_mouseEntered(MouseEvent e) {
  jButton1.setEnabled(false);
 }

 void jButton1_mouseExited(MouseEvent e) {
  jButton1.setEnabled(true);
 }

 void jButton2_mouseClicked(MouseEvent e) {
  jButton2.setText("No Help");
 }
}
```

I have purposely left the code "as is", so that we can be reminded of the quality of code generated by GUI builders.

I will show you some classes that do the same, but take less code and are more readable. First, we extract the functionality to centre a window on the screen into a common class. Since this will be shared between other parts of our system, we should not count this towards the lines of code that we need.

```
import java.awt.*;

public class Windows {
  public static void centerOnScreen(Window window) {
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    Dimension windowSize = window.getSize();
    windowSize.height = Math.min(windowSize.height, screenSize.height);
    windowSize.width = Math.min(windowSize.width, screenSize.width);
    window.setLocation((screenSize.width - windowSize.width) / 2,
      (screenSize.height - windowSize.height) / 2);
  }
}
```

Then, we use the refactoring built into Eclipse to make the frame a bit more human editable:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;

public class MultiplicationTable extends JFrame {
  private final JButton okButton = new JButton("OK");
  private final JButton helpButton = new JButton("Help");
  private final JTable table = new JTable(new DefaultTableModel(100, 20) {
    public String getColumnName(int column) {
      return Integer.toString(column+1);
    }
    public Object getValueAt(int row, int column) {
      return Integer.toString((row+1)*(column+1));
    }
  });

  //Construct the frame
  public MultiplicationTable() {
    super("Multiplication Table");
    setSize(603, 483);

    okButton.addMouseListener(new MouseAdapter() {
      public void mouseEntered(MouseEvent e) {
        okButton.setEnabled(false);
      }
      public void mouseExited(MouseEvent e) {
        okButton.setEnabled(true);
      }
    });
    helpButton.addMouseListener(new MouseAdapter() {
      public void mouseClicked(MouseEvent e) {
        helpButton.setText("No Help");
      }
    });
    JPanel titlePanel = new JPanel();
    titlePanel.add(new JLabel("Multiplication Table"));
    getContentPane().add(titlePanel, BorderLayout.NORTH);
    getContentPane().add(new JScrollPane(table), BorderLayout.CENTER);
    JPanel buttonPanel = new JPanel();
    buttonPanel.add(okButton);
    buttonPanel.add(helpButton);
    getContentPane().add(buttonPanel, BorderLayout.SOUTH);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }

  // instead of Application1, we have the following few lines
  public static void main(String[] args) {
    MultiplicationTable frame = new MultiplicationTable();
    Windows.centerOnScreen(frame);
    // instead of enabling the events and listening to window events:
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.show();
  }
}
```

This is better, in that the code has shrunk from 123 LOC down to just 54 lines. This is a simple example, with a complicated example, we would improve by even more. However, we have still not achieved very much reuse. We cannot use this frame in any other way except as a multiplication table.

Factory Method according to GoF

Seeing that you are subscribed to **The Java(tm) Specialists' Newsletter**, you are either a famous author of a Design Patterns book, or you have a copy of at least the Gang-of-Four book in your bookshelf. Yes? If not, I can highly recommend the book by Erich Gamma, et al.

The book contains a Design Pattern called the **Factory Method**. Believe it or not, but that pattern took me the longest to grasp when I developed my course on Design Patterns. I can therefore recommend that you read the pattern in the book until you do not understand it anymore, and then read it a few more times until you understand it again. That process is called "being humbled" and it is the only way that I know in which a human can learn Design Patterns. Only once you realise that you do not understand, can you open up your mind to learn.

The intent of **Factory Method** according to the GoF book is: *"Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses."*

Let us remember that when the book was written, Java had not been conceived and the examples were based on C++, which does not have interfaces. "Interface" therefore refers to the methods available in the class, it does *not* mean that the top-level class has to be an interface, a la Java.

The first step to make this example more extendable is to have creational methods for where we are creating objects:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;

public class MultiplicationTable2 extends JFrame {
  public MultiplicationTable2(String title) {
    super(title);
    JPanel titlePanel = new JPanel();
    titlePanel.add(getDescription());
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    getContentPane().add(titlePanel, BorderLayout.NORTH);

    JTable table = new JTable(makeTableModel());
    getContentPane().add(new JScrollPane(table), BorderLayout.CENTER);

    JPanel buttonPanel = new JPanel();
    JButton[] buttons = makeButtons();
    for (int i = 0; i < buttons.length; i++) {
      buttonPanel.add(buttons[i]);
    }
    getContentPane().add(buttonPanel, BorderLayout.SOUTH);
  }

  protected TableModel makeTableModel() {
    return new DefaultTableModel(100, 20) {
      public String getColumnName(int column) {
        return Integer.toString(column+1);
      }
      public Object getValueAt(int row, int column) {
        return Integer.toString((row+1)*(column+1));
      }
    };
  }

  protected JLabel getDescription() {
    return new JLabel("Multiplication Table");
  }
  protected JButton[] makeButtons() {
    final JButton okButton = new JButton("OK");
    final JButton helpButton = new JButton("Help");
    okButton.addMouseListener(new MouseAdapter() {
      public void mouseEntered(MouseEvent e) {
        okButton.setEnabled(false);
      }
      public void mouseExited(MouseEvent e) {
        okButton.setEnabled(true);
      }
    });
    helpButton.addMouseListener(new MouseAdapter() {
      public void mouseClicked(MouseEvent e) {
        helpButton.setText("No Help");
      }
    });
    return new JButton[] { okButton, helpButton };
  }

  public static void main(String[] args) {
    MultiplicationTable2 frame = new MultiplicationTable2("Multiplication Table");
    // it is better to set the size outside of the frame construction
    frame.setSize(603, 483);
    Windows.centerOnScreen(frame);
    // instead of enabling the events and listening to window events:
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.show();
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

}

This refactoring exercise has actually *increased* the code by 12 lines.

Let us assume that we will have several such frames with tables and buttons in our system. We can write an AbstractTableFrame class that contains **Factory Methods** which we can subclass to make our MultiplicationTable3:

```
import java.awt.*;
import javax.swing.*;
import javax.swing.table.*;

public abstract class AbstractTableFrame extends JFrame {
  public AbstractTableFrame(String title) {
    super(title);
    JPanel titlePanel = new JPanel();
    titlePanel.add(getDescription());
    getContentPane().add(titlePanel, BorderLayout.NORTH);

    JTable table = new JTable(makeTableModel());
    getContentPane().add(new JScrollPane(table), BorderLayout.CENTER);

    JPanel buttonPanel = new JPanel();
    JButton[] buttons = makeButtons();
    for (int i = 0; i < buttons.length; i++) {
      buttonPanel.add(buttons[i]);
    }
    getContentPane().add(buttonPanel, BorderLayout.SOUTH);
  }

  protected abstract TableModel makeTableModel();

  protected abstract JLabel getDescription();

  protected abstract JButton[] makeButtons();
}
```

We can now simply subclass this AbstractTableFrame and specify the **Model** and the **Controller** of the frame. The **View** is made by the AbstractTableFrame class. This separation of concerns is worth the effort that we have put into breaking up the original RAD generated code into these classes.

```
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;

public class MultiplicationTable3 extends AbstractTableFrame {
  public MultiplicationTable3() {
    super("Multiplication Table");
  }

  protected TableModel makeTableModel() {
    return new DefaultTableModel(100, 20) {
      public String getColumnName(int column) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      return Integer.toString(column+1);
    }
    public Object getValueAt(int row, int column) {
      return Integer.toString((row+1)*(column+1));
    }
  };
 }

 protected JLabel getDescription() {
  return new JLabel("Multiplication Table");
 }
 protected JButton[] makeButtons() {
  final JButton okButton = new JButton("OK");
  final JButton helpButton = new JButton("Help");
  okButton.addMouseListener(new MouseAdapter() {
    public void mouseEntered(MouseEvent e) {
      okButton.setEnabled(false);
    }
    public void mouseExited(MouseEvent e) {
      okButton.setEnabled(true);
    }
  });
  helpButton.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent e) {
      helpButton.setText("No Help");
    }
  });
  return new JButton[] { okButton, helpButton };
 }

 public static void main(String[] args) {
  MultiplicationTable3 frame = new MultiplicationTable3();
  frame.setSize(603, 483);
  Windows.centerOnScreen(frame);
  // instead of enabling the events and listening to window events:
  frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  frame.show();
 }
}
```

We do not achieve that much in terms of lines of code, since the MultiplicationTable3 class is only 3 lines shorter than MultiplicationTable. However, we have achieved the holy mantra of reusability, reusability, reusability. For example, let's write a new frame that contains a simple 3x4 table and one button:

```
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;

public class SimpleTableFrame extends AbstractTableFrame {
 public SimpleTableFrame() {
  super("Simple Table");
 }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  protected TableModel makeTableModel() {
    return new DefaultTableModel(3, 4);
  }

  protected JLabel getDescription() {
    return new JLabel("Empty Default Table");
  }

  protected JButton[] makeButtons() {
    return new JButton[] { new JButton(new AbstractAction("Close") {
      public void actionPerformed(ActionEvent e) {
        dispose();
      }
    })};
  }

  public static void main(String[] args) {
    SimpleTableFrame frame = new SimpleTableFrame();
    frame.setSize(603, 483);
    Windows.centerOnScreen(frame);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.show();
  }
}
```

We have managed to write this frame in just 33 lines of code, and we could have done it with Emacs, vi, notepad, even edlin!

What Have We Achieved?

Reducing the number of lines is just one of the benefits of applying the **Factory Method** Design Pattern. In addition, we have now got a company-wide frame that we can reuse to represent any frame containing a table. What is more, say we would like to change the **View** of this frame, we only have to change **one** class, and all other frames will also look different. Very powerful stuff indeed.

My recommendation is that you write all of your Java GUIs like this and that you look for opportunities where you can apply the Factory Method in other code. In the long run, it will make your code more manageable and maintainable. Unless of course you are a highly-paid contractor who is paid per hour. Then rather use the RAD tools because your boss will think you are working "rapidly". I expect a cut from your next invoice for that tip ;-)

I have to thank the folks at jGuru who many years ago pointed me in this direction and who made me rethink my views on RAD GUI development in Java.

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 075] - An Automatic Wait Cursor: WaitCursorEventQueue

Author: Nathan Arthur

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 75th edition of **The Java(tm) Specialists' Newsletter**. This week we are re-looking at the problem of wait cursors. I am very grateful to Nathan Arthur and his company for allowing us to publish this article. They have been through a lot of effort in producing this code, and I am sure it will help you as well. There are two things that I would like you to note in this article. First, it is practical, real-world code. There might be bugs, if you find any, please let us know. It is extremely useful, well-written code. Second, Nathan is passionate about unit testing. The more I work as a programmer, the more I feel the same passion as Nathan.

There are many ways to skin a cat. [I will not describe them in this newsletter, since that would be slightly off-topic.] In the same way, there are many waits t'askin a User to wait. [In the unlikely event that one of my old English teachers is reading this newsletter - the previous sentence construction was intentional.]

Enough of me, let us listen to what Nathan has to say (you at the back of the class, please be quiet and listen as well. Yes, **you**.) If you would like to send thanks and comments to Nathan, please email to truist-waitcursor@truist.com.

An Automatic Wait Cursor: WaitCursorEventQueue

A few months ago, I received the "Wait, Cursor, Wait!" edition of this newsletter. I had just spent man-weeks trying to discover better approaches to using wait cursors in my application, and I had two immediate reactions: First, it amused me that it came out only a few days after I had just done all that work. Second, it worried me that it might present a simpler solution than the one I found, thus making all the time I had spent a waste. Thankfully, it did not. It presented a solution very similar to the solution my application had *started* with. You will therefore benefit from my weeks of hard labour :-)

For some background, you should probably read issue # 16 ("Blocking Queue") and issue # 7 ("java.awt.EventQueue") of **The Java(tm) Specialists' Newsletter**. At the end of this article, we will have implemented our own EventQueue.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Analysis of "Wait, Cursor, Wait!"

Before I get into the background, I would like to discuss a few things I noted in "Wait, Cursor, Wait!"

First, the use of the GlassPane is not quite enough in that example - you will notice that you can still use the keyboard to tab around to components in the parent frame.

Second, Java modal dialogs on Windows are "natively" modal, which you can see by clicking on the frame behind a modal dialog - the title bar of the modal dialog will flash. This does not happen if the dialog is not modal, and there is no way to mimic the behavior in java.

Third, at the end of "Solution 1," there is a paragraph explaining that the strange wait cursor behavior is caused by the first event pump being blocked while the modal dialog is open. This is not actually the cause of the problem - the cause is explained in the Sun bug #4282540: the wait cursor does not paint on the frame because that is the correct Windows behavior for frames behind modal dialogs. Presumably, that is implemented in the frame's native peer. I would be interested to see what happens on other operating systems with different modal dialog / frame cursor behaviors.

Because of all these, I prefer to use actual modal dialogs, not use the glass pane for the wait cursor, and to live with the wait cursor not showing on the parent frame. I also would really prefer to have an automatic wait cursor solution, so I do not have to explicitly deal with it everywhere, and so the application always shows it even if I forgot to turn it on.

My Original Solution

The original wait cursor implementation I set out to improve was this:

```
import java.awt.*;
import java.awt.event.InputEvent;
import java.util.*;
import javax.swing.SwingUtilities;

public class WaitControl {
  private static int waiting = 0;
  private static ArrayList events = new ArrayList();
  private static final Cursor DEFAULT_CURSOR =
    Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR);
  private static final Cursor WAIT_CURSOR =
    Cursor.getPredefinedCursor(Cursor.WAIT_CURSOR);
  public static Window startWait(Component componentInWindow) {
    Window window = getEnclosingWindow(componentInWindow);
    if (window == null) {
      return null;
    }
    waiting++;
    // Only wait if we are not already
    if (waiting == 1) {
      window.setCursor(WAIT_CURSOR);
      EventQueue q = window.getToolkit().getSystemEventQueue();
      try {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      while (q.peekEvent() != null) {
        events.add(q.getNextEvent());
      }
    } catch (InterruptedException ex) {
      Thread.currentThread().interrupt();
    }
  }
  return window;
}
public static void endWait(Component componentInWindow) {
  Window window = getEnclosingWindow(componentInWindow);
  if (window == null) {
    return;
  }
  if (waiting > 0) {
    waiting--;
    // Only stop when all waiting is done
    if (waiting == 0) {
      EventQueue q = window.getToolkit().getSystemEventQueue();
      try {
        while (q.peekEvent() != null) {
          AWTEvent event = q.getNextEvent();
          if (!(event instanceof InputEvent)) {
            events.add(event);
          }
        }
      } catch (InterruptedException ex) {
        Thread.currentThread().interrupt();
      }
      for (Iterator it = events.iterator(); it.hasNext();) {
        q.postEvent((AWTEvent) it.next());
      }
      window.setCursor(DEFAULT_CURSOR);
      events.clear();
    }
  }
}
public static void fullEndWait(Window window) {
  if (waiting > 0) {
    waiting = 1;
    endWait(window);
  }
}
public static Window getEnclosingWindow(Component componentInWindow) {
  if (componentInWindow instanceof Window) {
    return (Window) componentInWindow;
  } else if (componentInWindow != null) {
    return SwingUtilities.windowForComponent(componentInWindow);
  } else {
    return null;
  }
}
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

At a high level, this implementation is simple. It filters input events by taking them directly off the system event queue, which means we do not have to use the GlassPane, and can use modal dialogs.

One of the many problems with this solution was that we always had to remember to call endWait() with the correct component and the right number of times. Ideally we would have preferred for the wait cursor to appear (and disappear) automagically.

The First EventQueue-based Attempt

Also, as it happened, we though we had found a way to make it work automatically. We found it online, in an article on JavaWorld.com, titled "Automate the hourglass cursor." There were some problems with that approach:

The first flaw is subtle does not cause much noticeable bad behavior. The problem lies in the assumption made about how wait() and interrupt() work together. Specifically, line 51 of the code assumes that if interrupt() is called during the wait(), an InterruptedException will be thrown. However, the InterruptedException will only be thrown if the delay timeout has not ended yet. If the timeout has ended, but the thread is still in contention for the object's monitor, interrupt() will simply set the interrupt status of the thread to **true**, without throwing the exception.

If you install the code as written, and use the application, you will occasionally see little flashes of the wait cursor that do not make sense. In addition, if you run the unit test at the end of the article, nearly all the tests will fail.

The flaw is fixable, however. Simply add these two lines of code after line 51:

```
if (Thread.interrupted())
  continue;
```

This performs essentially the same function as the exception would have, and fixes the problem. Note that we use Thread.interrupted() and not isInterrupted() because Thread.interrupted() resets the flag, while isInterrupted() does not.

Another problem is that modal dialogs cause the dispatchEvent() method to be recursive. I will explain why this is true below, but for the moment, just take my word for it - opening a modal dialog in the call to super.dispatchEvent() will cause dispatchEvent() to get called again before super.dispatchEvent() returns. This causes a problem.

The problem is fairly obvious - two calls to waitTimer.startTimer() in a row, without an intervening call to waitTimer.stopTimer(), will cause both wait() calls in the run() loop to be passed, resulting in a wait cursor being set on the dialog, and the state tracking to be broken. As a user, you will not notice this much, because usually the cursor will get immediately reset, but this does cause problems if the code that was showing the dialog takes a long time after the dialog is closed. No cursor will get set, because the stopTimer() call from the first call to dispatchEvent will then execute, canceling any cursor that would have otherwise been set. In addition, the tracking is at this point out of sync, and can cause further problems (although it usually resets itself correctly).

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

This is a common case - imagine starting some long operation, at the beginning of which you open a dialog to ask a user to make a choice. In this case, the long operation will not (reliably) have a wait cursor, because of this bug in the code. The flaw is not simple to fix, and requires a detailed understanding of how modal dialogs and the event queue mechanism work.

The System Event Queue, System Event Pump, and Modal Dialogs

Since this is **The Java(tm) Specialists' Newsletter**, I can probably assume that you have a basic understanding of the system event queue. However, I have found that very few Java developers have a clear idea of what happens when a modal dialog is opened.

First, all events generated by the user (mouse, keyboard, system, etc.) are passed through the VM, and all program events (paints, actions, etc.) are placed on the system event queue. (Note that there is usually exactly one event queue per program, but sometimes an event queue will be shared across applets.) These events are generated asynchronously, and are simply placed on the queue from whichever thread they were generated in. This is happening for the entire life of the AWT program.

Second, and at the same time, there is a special thread (commonly called the "AWT event thread" and named "AWT-EventQueue-0") that is constantly removing events (in FIFO order) from the event queue, and dispatching them. Note that the currently dispatching event is not on the queue - it is removed before it is dispatched. All GUI operations should be performed on this thread, because the AWT and Swing code is not thread safe. There is always exactly one active event thread per event queue.

Third, when a modal dialog is shown, a new event pump is (normally) started, which takes over for the previous event pump and starts pulling events off the event queue and dispatching them. This new pump shuts down when the modal dialog is closed.

The confusion usually happens because people believe that starting a new pump means starting a new thread, but that is not true. (In fact, it would not work.) What really happens is that Dialog.show() simply takes over the job of pumping the event queue. This is why calling setVisible() on a modal dialog does not return until the dialog is closed - setVisible() continues pumping events.

There is only ever one active event thread. When a modal dialog is opened, this happens as part of an event on the event thread, and that event simply does not quit, and starts pumping events so that events continue to happen.

This explains why EventQueue.dispatchEvent() is recursive with modal dialogs - the dialog itself will start pumping events (and call dispatchEvent()) before the event that showed it returns.

If you are interested in the details of all this, I would suggest reading the code. Start with java.awt.EventQueue, and pay attention to postEvent(), postEventPrivate(), and the other version of postEvent(). Pay particular attention to getNextEvent() and dispatchEvent(). Also, look at the push() and pop() methods.

Then look at java.awt.EventDispatchThread, specifically at run(), pumpEvents(), the other pumpEvents(), pumpEventsForHierarchy(), and pumpOneEventForHierarchy().

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

I do not really understand why this code filters events for the modal dialog, because there should not ever be events outside of the modal dialog, but perhaps it has some use on another platform. If somebody knows, I would love to hear it!

Finally, look at java.awt.Dialog, specifically at show(). That will bring it all together.

A New WaitCursorEventQueue

We have now seen three implementations of a wait cursor manager - the one presented in issue #65, the one presented at the top of this document, and the one on JavaWorld.com. None of them is perfect, but they all have some good features. We will take the best from each of them, and add a few improvements of our own. Specifically, we want to build a cursor manager that:

- Automatically displays and resets the cursor, remembering the original cursor
- Works with modal dialogs (recursive calls to dispatchEvent(), and correct filtering of events)
- Correctly displays the cursor for all long events, even after a modal dialog is closed
- Is thread safe where it needs to be does not impose a significant performance overhead
- Is thoroughly unit tested

The final implementation of these "requirements" relies on four classes and an interface, and two unit tests. The primary class is WaitCursorEventQueue, and it works with a CursorManager class, which uses a DispatchedEvent class. The remaining class is DelayTimer, which, with the DelayTimerCallback interface, implements a generic delay timer that can be used for other purposes, if needed.

DelayTimer

I first present the DelayTimer. It is thread safe, and supports any series of calls to startTimer() and stopTimer().

[HK: I personally prefer to use the interrupted flag to indicate that I want to quit a thread. However, I did not want to break anything, so here is the code as it was :-]

```
/**
 * This class implements a delay timer that will call trigger()
 * on the DelayTimerCallback delay milliseconds after
 * startTimer() was called, if stopTimer() was not called first.
 * The timer will only throw events after startTimer() is called.
 * Until then, it does nothing.  It is safe to call stopTimer()
 * and startTimer() repeatedly.
 *
 * Note that calls to trigger() will happen on the timer thread.
 *
 * This class is multiple-thread safe.
 */
public class DelayTimer extends Thread {
  private final DelayTimerCallback callback;
  private final Object mutex = new Object();
  private final Object triggeredMutex = new Object();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  private final long delay;
  private boolean quit;
  private boolean triggered;
  private long waitTime;

  public DelayTimer(DelayTimerCallback callback, long delay) {
   this.callback = callback;
   this.delay = delay;
   setDaemon(true);
   start();
  }
  /**
   * Calling this method twice will reset the timer.
   */
  public void startTimer() {
   synchronized (mutex) {
    waitTime = delay;
    mutex.notify();
   }
  }
  public void stopTimer() {
   try {
    synchronized (mutex) {
     synchronized (triggeredMutex) {
      if (triggered) {
       triggeredMutex.wait();
      }
     }
     waitTime = 0;
     mutex.notify();
    }
   } catch (InterruptedException ie) {
    System.err.println("trigger failure");
    ie.printStackTrace(System.err);
   }
  }
  public void run() {
   try {
    while (!quit) {
     synchronized (mutex) {
      //we rely on wait(0) being implemented to wait forever here
      if (waitTime < 0) {
       triggered = true;
       waitTime = 0;
      } else {
       long saveWaitTime = waitTime;
       waitTime = -1;
       mutex.wait(saveWaitTime);
      }
     }
     try {
      if (triggered) {
       callback.trigger();
      }
     } catch (Exception e) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
        System.err.println(
            "trigger() threw exception, continuing");
        e.printStackTrace(System.err);
      } finally {
        synchronized (triggeredMutex) {
          triggered = false;
          triggeredMutex.notify();
        }
      }
    }
  } catch (InterruptedException ie) {
    System.err.println("interrupted in run");
    ie.printStackTrace(System.err);
  }
 }
 public void quit() {
  synchronized (mutex) {
    this.quit = true;
    mutex.notify();
  }
 }
}
```

The DelayTimer relies on DelayTimerCallback:

```
public interface DelayTimerCallback {
public void trigger();
}
```

We rely on the correct functioning of this class for the WaitCursorEventQueue, so it is very important that it have a unit test:

```
import junit.framework.TestCase;

public class DelayTimerTest extends TestCase {
  private static final int TIMEOUT = 100;
  private static final int BUFFER = 20;
  private static final int MORE_THAN_HALF = 60;
  private DelayTimer timer;
  private TestDelayTimerCallback callback;

  public DelayTimerTest(String name) {
    super(name);
  }

  public void setUp() {
    callback = new TestDelayTimerCallback();
    timer = new DelayTimer(callback, TIMEOUT);
  }

  public void tearDown() {
    timer.quit();
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
 private void sleep(long millis) {
   try {
    Thread.sleep(millis);
   } catch (InterruptedException e) {
    e.printStackTrace();
   }
 }

 public void testNotStarted() {
   sleep(TIMEOUT + BUFFER);
   assertEquals("no trigger without start()", 0,
            callback.getTriggerCount());
 }

 public void testNoTriggerIfTooShort() {
   timer.startTimer();
   assertEquals("no trigger if too fast", 0,
            callback.getTriggerCount());
   timer.stopTimer();
   sleep(TIMEOUT + BUFFER);
   assertEquals("no trigger after stop", 0,
            callback.getTriggerCount());
 }

 public void testTimerRestarts() {
   timer.startTimer();
   sleep(MORE_THAN_HALF);
   timer.startTimer();
   sleep(MORE_THAN_HALF);
   timer.stopTimer();
   assertEquals("timer is restarted on calls to start()", 0,
            callback.getTriggerCount());
 }

 public void testTimerTriggersThenStops() {
   timer.startTimer();
   sleep(TIMEOUT + BUFFER);
   timer.stopTimer();
   sleep(TIMEOUT + BUFFER);
   assertEquals("timer triggered event", 1,
            callback.getTriggerCount());
 }

 public void testTimerOnlyTriggersOneEvent() {
   timer.startTimer();
   sleep(TIMEOUT + BUFFER);
   assertEquals("timer triggered event", 1,
            callback.getTriggerCount());
   sleep(TIMEOUT + BUFFER);
   assertEquals("timer did not trigger another event", 1,
            callback.getTriggerCount());
   timer.stopTimer();
 }

 public void testTimerStopsTwice() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    timer.startTimer();
    timer.stopTimer();
    timer.stopTimer();
    sleep(TIMEOUT + BUFFER);
    assertEquals("timer did not trigger event", 0,
                 callback.getTriggerCount());
  }

  public void testTriggers() {
    timer.startTimer();
    sleep(TIMEOUT + BUFFER);
    timer.stopTimer();
    timer.stopTimer();
    assertEquals("timer triggered only 1 event", 1,
                 callback.getTriggerCount());
  }

  public void testTriggerTrigger() {
    timer.startTimer();
    sleep(TIMEOUT + BUFFER);
    assertEquals("timer triggered first event", 1,
                 callback.getTriggerCount());
    timer.startTimer();
    sleep(TIMEOUT + BUFFER);
    assertEquals("timer triggered second event", 2,
                 callback.getTriggerCount());
    sleep(TIMEOUT + BUFFER);
    timer.stopTimer();
    assertEquals("timer did not trigger another event", 2,
                 callback.getTriggerCount());
  }

  public void testStopTimerHappensAfterTrigger() {
    FancyTestDelayTimerCallback callback = new FancyTestDelayTimerCallback();
    timer = new DelayTimer(callback, TIMEOUT);

    timer.startTimer();
    sleep(TIMEOUT + BUFFER);
    assertTrue("timer is in trigger", callback.inTrigger);
    assertTrue("timer has not thrown exception",
               !callback.exception);
    assertTrue("timer is not out of trigger",
               !callback.outOfTrigger);

    Runnable runnable = new Runnable() {
      public void run() {
        timer.stopTimer();
      }
    };

    Thread testThread = new Thread(runnable, "test thread");
    testThread.start();

    sleep(TIMEOUT + BUFFER);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    assertTrue("stopTimer() has not returned",
            testThread.isAlive());
    synchronized (callback) {
      callback.notify();
    }

    sleep(TIMEOUT + BUFFER);

    assertTrue("timer has not thrown exception",
            !callback.exception);
    assertTrue("timer is out of trigger", callback.outOfTrigger);
    assertTrue("stopTimer() has returned", !testThread.isAlive());
  }

  public void testMishMash() {
    timer.startTimer();
    sleep(MORE_THAN_HALF);
    timer.startTimer();
    sleep(MORE_THAN_HALF);
    timer.stopTimer();
    sleep(MORE_THAN_HALF);
    timer.startTimer();
    timer.stopTimer();
    timer.stopTimer();
    sleep(MORE_THAN_HALF);
    timer.startTimer();
    timer.startTimer();
    assertEquals("no event yet", 0, callback.getTriggerCount());
    sleep(TIMEOUT + BUFFER);
    timer.stopTimer();
    assertEquals("got event yet", 1, callback.getTriggerCount());
  }

  private class FancyTestDelayTimerCallback
      implements DelayTimerCallback {
    public boolean exception;
    public boolean inTrigger;
    public boolean outOfTrigger;

    public synchronized void trigger() {
      inTrigger = true;

      try {
        wait();
      } catch (InterruptedException e) {
        exception = true;
        e.printStackTrace();
      } finally {
        outOfTrigger = true;
      }
    }
  }

  private class TestDelayTimerCallback
      implements DelayTimerCallback {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    private int triggerCount;

    public int getTriggerCount() {
      return triggerCount;
    }

    public void trigger() {
      triggerCount++;
    }
  }
}
```

WaitCursorEventQueue

Now that we have seen the DelayTimer, I am going to skip ahead now to WaitCursorEventQueue itself, which is a fairly simple class:

```
import java.awt.*;
/**
 * Suggested serving size:
 * Toolkit.getDefaultToolkit().getSystemEventQueue().push(new
WaitCursorEventQueue(70));
 */
public class WaitCursorEventQueue extends EventQueue
    implements DelayTimerCallback {
  private final CursorManager cursorManager;
  private final DelayTimer waitTimer;

  public WaitCursorEventQueue(int delay) {
    this.waitTimer = new DelayTimer(this, delay);
    this.cursorManager = new CursorManager(waitTimer);
  }
  public void close() {
    waitTimer.quit();
    pop();
  }
  protected void dispatchEvent(AWTEvent event) {
    cursorManager.push(event.getSource());
    waitTimer.startTimer();
    try {
      super.dispatchEvent(event);
    } finally {
      waitTimer.stopTimer();
      cursorManager.pop();
    }
  }
  public AWTEvent getNextEvent() throws InterruptedException {
    waitTimer.stopTimer(); //started by pop(), this catches modal dialogs
    //closing that do work afterwards
    return super.getNextEvent();
  }
  public void trigger() {
    cursorManager.setCursor();
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

The most significant method in this class is dispatchEvent(). CursorManager manages a stack of events (and their sources) so it can handle modal dialogs and set the cursor on the right window.

If a modal dialog opened during the call to super.dispatchEvent(), we will get another call to dispatchEvent(). This will tell the CursorManager that a new event is started and call startTimer() again. We rely on the fact that a second call to startTimer() will restart the timer, resetting the delay. (This way, a modal dialog does not immediately get a wait cursor set.) Once this new event is over, we will stop the timer, which stops it completely, even though startTimer() was called twice. The CursorManager will still have the event that opened the modal dialog on its stack, which is correct, because that event has not finished yet. Once the modal dialog is closed, that event will finish, and take the event off the CursorManager's stack.

CursorManager

The CursorManager class might be more appropriately named "WindowManager" or "EventManager", but none of those names are quite right either, and CursorManager is what it started as, so I will leave it that way. This class' primary job is to assist the WaitCursorEventQueue in managing the events, by handling modal dialogs (recursiveness), filtering events, and helping to manage the WaitTimer. Here is the code:

```
import java.awt.*;
import java.awt.event.InputEvent;
import java.util.*;

class CursorManager {
  private final DelayTimer waitTimer;
  private final Stack dispatchedEvents;
  private boolean needsCleanup;

  public CursorManager(DelayTimer waitTimer) {
    this.dispatchedEvents = new Stack();
    this.waitTimer = waitTimer;
  }
  private void cleanUp() {
    if (((DispatchedEvent) dispatchedEvents.peek()).resetCursor()) {
      clearQueueOfInputEvents();
    }
  }
  private void clearQueueOfInputEvents() {
    EventQueue q = Toolkit.getDefaultToolkit().getSystemEventQueue();
    synchronized (q) {
      ArrayList nonInputEvents = gatherNonInputEvents(q);
      for (Iterator it = nonInputEvents.iterator(); it.hasNext();)
        q.postEvent((AWTEvent)it.next());
    }
  }
  private ArrayList gatherNonInputEvents(EventQueue systemQueue) {
    ArrayList events = new ArrayList();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    while (systemQueue.peekEvent() != null) {
      try {
        AWTEvent nextEvent = systemQueue.getNextEvent();
        if (!(nextEvent instanceof InputEvent)) {
          events.add(nextEvent);
        }
      } catch (InterruptedException ie) {
        Thread.currentThread().interrupt();
      }
    }
    return events;
  }
  public void push(Object source) {
    if (needsCleanup) {
      waitTimer.stopTimer();
      cleanUp(); //this corrects the state when a modal dialog
      //opened last time round
    }
    dispatchedEvents.push(new DispatchedEvent(source));
    needsCleanup = true;
  }
  public void pop() {
    cleanUp();
    dispatchedEvents.pop();
    if (!dispatchedEvents.isEmpty()) {
      //this will be stopped if getNextEvent() is called -
      //used to watch for modal dialogs closing
      waitTimer.startTimer();
    } else {
      needsCleanup = false;
    }
  }
  public void setCursor() {
    ((DispatchedEvent) dispatchedEvents.peek()).setCursor();
  }
}
```

Most of the code is self-explanatory. Only pop() is a bit complicated. Once it has unset the cursor and popped the event, it checks to see if there are any outstanding events on the stack. If so, it knows that there must be a modal dialog open and needs to take special precautions.

Yes, it is a bit heavy-handed to start the timer on every single event when a modal dialog is up - but it works, and it does not actually impose much overhead.

DispatchedEvent

The last interesting class is DispatchedEvent. If you recall, we already know that it represents an event on the EventQueue, and that it can set and unset the cursor on the appropriate window.

```
import java.awt.*;
import javax.swing.SwingUtilities;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
class DispatchedEvent {
  private final Object mutex = new Object();
  private final Object source;
  private Component parent;
  private Cursor lastCursor;

  public DispatchedEvent(Object source) {
    this.source = source;
  }
  public void setCursor() {
    synchronized (mutex) {
      parent = findVisibleParent();
      if (parent != null) {
        lastCursor = (parent.isCursorSet() ? parent.getCursor() : null);
        parent.setCursor(Cursor.getPredefinedCursor(Cursor.WAIT_CURSOR));
      }
    }
  }
  public boolean resetCursor() {
    synchronized (mutex) {
      if (parent != null) {
        parent.setCursor(lastCursor);
        parent = null;
        return true;
      }
      return false;
    }
  }
  private Component findVisibleParent() {
    Component result = null;
    if (source instanceof Component) {
      result = SwingUtilities.getRoot((Component) source);
    } else if (source instanceof MenuComponent) {
      MenuContainer mParent = ((MenuComponent) source).getParent();
      if (mParent instanceof Component) {
        result = SwingUtilities.getRoot((Component) mParent);
      }
    }
    if ((result != null) && result.isVisible()) {
      return result;
    } else {
      return null;
    }
  }
}
```

WaitCursorEventQueueTest

We have finally gotten through all the code, but how do we know it works? That is where unit testing comes in. Some people argue that you cannot unit test multi-threaded code, but I disagree - you can unit test it, so long as you are willing to live with an occasional false negative. What you do is write the test so that if there is a bug in the algorithm, the unit test will show it. This can be done by writing tests that

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

will pass only if all the threads line up the way you want them to, and all the events happen in the right order and right way.

If you have written a unit test that way then a run of the unit test might go three possible ways. First, it might hang, in which case you know you know there is a bug. Sometimes tracking it down can be extremely difficult, but it can be done. Second, the test might fail. In that case, it is usually a good idea to rerun the test and see if it fails repeatedly. If there really is something broken in the code, then the test should fail consistently. If it does not always fail, then you have to analyze why it is sometimes failing. If there is no bug, but just a race condition in the test, ignore it. If you find a bug, you can write a test that will reliably demonstrate it, and then fix the problem. Finally, all the tests might pass. If that is the case, then you either have working code or not enough tests. If you are not sure if you have enough tests, try going through all the code commenting out one line at a time, and running the tests. Every single commented out line should cause a repeatable failure. *[HK: I particularly like the last two sentences, excluding my own. Ok, let us include mine. Make it the last five sentences. :-]*

This unit test was designed and built with all of this in mind. Commenting out lines of code causes a test to fail. The tests do occasionally fail unreliably, but every time that has happened, I have been able to understand why. They have never locked up. Finally, two of the tests exist because of bugs found in the initial implementation of this code that the unit tests were occasionally catching.

Note also that this test relies on two timing parameters - how long the delay should be for a trigger, and how long an event should take to ensure that the trigger would have happened. Both of these are configurable constants at the top of the test, and You will probably need to adjust the numbers to your machine/OS/VM.

```
import java.awt.*;
import java.awt.event.*;
import java.lang.reflect.InvocationTargetException;
import javax.swing.*;
import junit.framework.TestCase;

public class WaitCursorEventQueueTest extends TestCase {
  private static final int TIMEOUT = 200;
  private static final int BUFFER = 30;
  private static final Cursor BASE_CURSOR = Cursor.getPredefinedCursor(
                                    Cursor.CROSSHAIR_CURSOR);
  private CursorReportingDialog dialog;
  private CursorReportingDialog dialog2;
  private CursorReportingFrame frame;
  private TestWaitCursorEventQueue eventQueue;

  public WaitCursorEventQueueTest(String name) {
    super(name);
  }

  public void setUp() {
    eventQueue = new TestWaitCursorEventQueue(TIMEOUT);
    Toolkit.getDefaultToolkit().getSystemEventQueue()
      .push(eventQueue);
    frame = new CursorReportingFrame();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    frame.pack();
    frame.setBounds(-1000, -1000, 100, 100);
    frame.setVisible(true);
    dialog = new CursorReportingDialog(frame);
    dialog.pack();
    dialog.setBounds(-1000, -1000, 100, 100);
    dialog2 = new CursorReportingDialog(dialog);
    dialog2.pack();
    dialog2.setBounds(-1000, -1000, 100, 100);
  }
  public void tearDown() throws InvocationTargetException,
                        InterruptedException {
    flushQueue();
    eventQueue.close();
    eventQueue = null;
    flushQueue();
    frame.dispose();
    frame = null;
    dialog.dispose();
    dialog = null;
  }
  private void flushQueue() throws InvocationTargetException,
                           InterruptedException {
    SwingUtilities.invokeAndWait(new Runnable() {
      public void run() {
      }
    });
  }
  private void postEvent(Object source, Runnable event) {
    eventQueue.postEvent(new InvocationEvent(source, event));
  }
  private void hangOut(long timeout) {
    try {
      Thread.sleep(timeout);
    } catch (InterruptedException e) {
      e.printStackTrace();
    }
  }
  public void testNoCursor() throws InvocationTargetException,
                            InterruptedException {
    DelayEvent event = new DelayEvent(TIMEOUT - BUFFER);
    postEvent(frame, event);
    postEvent(frame, event);
    postEvent(frame, event);
    postEvent(frame, event);
    flushQueue();
    assertEquals("no cursor set", 0, frame.getCursorSetCount());
    assertEquals("no cursor reset", 0,
                 frame.getCursorResetCount());
  }
  public void testCursor() throws InvocationTargetException,
                           InterruptedException {
    DelayEvent event = new DelayEvent(TIMEOUT + BUFFER);
    postEvent(frame, event);
    flushQueue();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    assertEquals("1 cursor set", 1, frame.getCursorSetCount());
    assertEquals("1 cursor reset", 1,
                 frame.getCursorResetCount());
  }
  public void testDialog() throws InvocationTargetException,
                                  InterruptedException {
    postEvent(frame, new DialogShowEvent(dialog, true, 0));
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog never got cursor", 0,
                 dialog.getCursorSetCount());
    assertEquals("dialog never reset cursor", 0,
                 dialog.getCursorResetCount());
    assertEquals("frame never got cursor", 0,
                 frame.getCursorSetCount());
    assertEquals("frame never reset cursor", 0,
                 frame.getCursorResetCount());
    postEvent(dialog, new DialogShowEvent(dialog, false, 0));
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog never got cursor", 0,
                 dialog.getCursorSetCount());
    assertEquals("dialog never reset cursor", 0,
                 dialog.getCursorResetCount());
    assertEquals("frame never got cursor", 0,
                 frame.getCursorSetCount());
    assertEquals("frame never reset cursor", 0,
                 frame.getCursorResetCount());
  }
  public void testCursorAndDialog()
                      throws InvocationTargetException,
                             InterruptedException {
    TestRunnable testAndShow = new TestRunnable() {
      public void run() {
        testsPassed &= (1 == frame.getCursorSetCount());
        testsPassed &= (0 == frame.getCursorResetCount());
        dialog.setVisible(true);
      }
    };
    postEvent(frame,
              new DelayEvent(TIMEOUT + BUFFER, testAndShow));
    flushQueue();
    assertTrue("Delay worked", testAndShow.getTestsPassed());
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog never got cursor", 0,
                 dialog.getCursorSetCount());
    assertEquals("dialog never reset cursor", 0,
                 dialog.getCursorResetCount());
    assertEquals("frame never got another cursor", 1,
                 frame.getCursorSetCount());
    assertEquals("frame reset cursor", 1,
                 frame.getCursorResetCount());
    postEvent(dialog, new DialogShowEvent(dialog, false, 0));
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    assertEquals("dialog never got cursor", 0,
                 dialog.getCursorSetCount());
    assertEquals("dialog never reset cursor", 0,
                 dialog.getCursorResetCount());
    assertEquals("frame never got another cursor", 1,
                 frame.getCursorSetCount());
    assertEquals("frame never reset another cursor", 1,
                 frame.getCursorResetCount());
  }
  public void testCursorAndDialogAndCursor()
    throws InvocationTargetException, InterruptedException {
    TestRunnable testAndShow = new TestRunnable() {
      public void run() {
        testsPassed &= (1 == frame.getCursorSetCount());
        testsPassed &= (0 == frame.getCursorResetCount());
        dialog.setVisible(true);
        hangOut(TIMEOUT + BUFFER);
      }
    };
    postEvent(frame,
              new DelayEvent(TIMEOUT + BUFFER, testAndShow));
    flushQueue();
    assertTrue("Delay worked", testAndShow.getTestsPassed());
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog never got cursor", 0,
                 dialog.getCursorSetCount());
    assertEquals("dialog never reset cursor", 0,
                 dialog.getCursorResetCount());
    assertEquals("frame never got another cursor", 1,
                 frame.getCursorSetCount());
    assertEquals("frame reset cursor", 1,
                 frame.getCursorResetCount());
    postEvent(dialog, new DialogShowEvent(dialog, false, 0));
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog never got cursor", 0,
                 dialog.getCursorSetCount());
    assertEquals("dialog never reset cursor", 0,
                 dialog.getCursorResetCount());
    assertEquals("frame got another cursor", 2,
                 frame.getCursorSetCount());
    assertEquals("frame reset another cursor", 2,
                 frame.getCursorResetCount());
  }
  /**
   * This test checks the condition where the EventDispatchThread
   * does not call EventQueue.getNextEvent() within TIMEOUT,
   * (presumably) because of thread contention, even when there
   * is not a dialog going down. Note that this case only actually
   * matters if there is a dialog currently up.
   */
  public void testDelayedGetNextEvent()
                        throws InvocationTargetException,
                               InterruptedException {
    postEvent(frame, new DialogShowEvent(dialog, true, 0));
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
    eventQueue.setGetDelay(TIMEOUT + BUFFER);
    postEvent(frame, new DelayEvent(TIMEOUT - BUFFER));
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
    assertEquals("frame got a cursor", 1,
                 frame.getCursorSetCount());
    assertEquals("frame reset a cursor", 1,
                 frame.getCursorResetCount());
    postEvent(dialog, new DialogShowEvent(dialog, false, 0));
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
    assertEquals("frame did not get another cursor", 1,
                 frame.getCursorSetCount());
    assertEquals("frame did not reset another cursor", 1,
                 frame.getCursorResetCount());
  }
  public void testTwoDialogs() throws InvocationTargetException,
                                      InterruptedException {
    TestRunnable testAndShow = new TestRunnable() {
      public void run() {
        testsPassed &= (1 == frame.getCursorSetCount());
        testsPassed &= (0 == frame.getCursorResetCount());
        dialog.setVisible(true);
      }
    };
    postEvent(frame,
              new DelayEvent(TIMEOUT + BUFFER, testAndShow));
    flushQueue();
    assertTrue("Delay worked", testAndShow.getTestsPassed());
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog never got cursor", 0,
                 dialog.getCursorSetCount());
    assertEquals("dialog never reset cursor", 0,
                 dialog.getCursorResetCount());
    assertEquals("frame never got another cursor", 1,
                 frame.getCursorSetCount());
    assertEquals("frame reset cursor", 1,
                 frame.getCursorResetCount());
    TestRunnable testAndShow2 = new TestRunnable() {
      public void run() {
        testsPassed &= (1 == dialog.getCursorSetCount());
        testsPassed &= (0 == dialog.getCursorResetCount());
        dialog2.setVisible(true);
      }
    };
    postEvent(dialog,
              new DelayEvent(TIMEOUT + BUFFER, testAndShow2));
    flushQueue();
    assertTrue("Delay worked", testAndShow.getTestsPassed());
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog2 never got cursor", 0,
                 dialog2.getCursorSetCount());
    assertEquals("dialog2 never reset cursor", 0,
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
            dialog2.getCursorResetCount());
    assertEquals("dialog never got another cursor", 1,
            dialog.getCursorSetCount());
    assertEquals("dialog reset cursor", 1,
            dialog.getCursorResetCount());
    assertEquals("frame never got another cursor", 1,
            frame.getCursorSetCount());
    assertEquals("frame reset cursor", 1,
            frame.getCursorResetCount());
    postEvent(dialog2, new DelayEvent(TIMEOUT + BUFFER));
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog2 got cursor", 1,
            dialog2.getCursorSetCount());
    assertEquals("dialog2 reset cursor", 1,
            dialog2.getCursorResetCount());
    assertEquals("dialog never got another cursor", 1,
            dialog.getCursorSetCount());
    assertEquals("dialog reset cursor", 1,
            dialog.getCursorResetCount());
    assertEquals("frame never got another cursor", 1,
             frame.getCursorSetCount());
    assertEquals("frame reset cursor", 1,
            frame.getCursorResetCount());
    postEvent(dialog2, new DialogShowEvent(dialog2, false, 0));
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog2 never got another cursor", 1,
            dialog2.getCursorSetCount());
    assertEquals("dialog2 never reset another cursor", 1,
            dialog2.getCursorResetCount());
    assertEquals("dialog never got another cursor", 1,
            dialog.getCursorSetCount());
    assertEquals("dialog never reset another cursor", 1,
            dialog.getCursorResetCount());
    assertEquals("frame never got another cursor", 1,
            frame.getCursorSetCount());
    assertEquals("frame never reset another cursor", 1,
            frame.getCursorResetCount());
    postEvent(dialog, new DialogShowEvent(dialog, false, 0));
    flushQueue();
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog2 never got another cursor", 1,
            dialog2.getCursorSetCount());
    assertEquals("dialog2 never reset another cursor", 1,
            dialog2.getCursorResetCount());
    assertEquals("dialog never got another cursor", 1,
            dialog.getCursorSetCount());
    assertEquals("dialog never reset another cursor", 1,
            dialog.getCursorResetCount());
    assertEquals("frame never got another cursor", 1,
            frame.getCursorSetCount());
    assertEquals("frame never reset another cursor", 1,
            frame.getCursorResetCount());
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public void testCursorAndTwoDialogsAndCursor()
  throws InvocationTargetException, InterruptedException {
  postEvent(frame, new DialogShowEvent(dialog, true, 0));
  flushQueue();
  hangOut(TIMEOUT + BUFFER);
  assertEquals("dialog never got cursor", 0,
               dialog.getCursorSetCount());
  assertEquals("dialog never reset cursor", 0,
               dialog.getCursorResetCount());
  assertEquals("frame never got cursor", 0,
               frame.getCursorSetCount());
  assertEquals("frame never reset cursor", 0,
               frame.getCursorResetCount());
  TestRunnable testAndShow = new TestRunnable() {
    public void run() {
      testsPassed &= (1 == dialog.getCursorSetCount());
      testsPassed &= (0 == dialog.getCursorResetCount());
      dialog2.setVisible(true);
      hangOut(TIMEOUT + BUFFER);
    }
  };
  postEvent(dialog,
            new DelayEvent(TIMEOUT + BUFFER, testAndShow));
  flushQueue();
  assertTrue("Delay worked", testAndShow.getTestsPassed());
  hangOut(TIMEOUT + BUFFER);
  assertEquals("dialog2 never got cursor", 0,
               dialog2.getCursorSetCount());
  assertEquals("dialog2 never reset cursor", 0,
               dialog2.getCursorResetCount());
  assertEquals("dialog never got another cursor", 1,
               dialog.getCursorSetCount());
  assertEquals("dialog reset cursor", 1,
               dialog.getCursorResetCount());
  assertEquals("frame never got cursor", 0,
               frame.getCursorSetCount());
  assertEquals("frame never reset cursor", 0,
               frame.getCursorResetCount());
  postEvent(dialog2, new DialogShowEvent(dialog2, false, 0));
  flushQueue();
  hangOut(TIMEOUT + BUFFER);
  assertEquals("dialog2 never got cursor", 0,
               dialog2.getCursorSetCount());
  assertEquals("dialog2 never reset cursor", 0,
               dialog2.getCursorResetCount());
  assertEquals("dialog got another cursor", 2,
               dialog.getCursorSetCount());
  assertEquals("dialog reset another cursor", 2,
               dialog.getCursorResetCount());
  assertEquals("frame never got cursor", 0,
               frame.getCursorSetCount());
  assertEquals("frame never reset cursor", 0,
               frame.getCursorResetCount());
  postEvent(dialog, new DialogShowEvent(dialog, false, 0));
  flushQueue();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    hangOut(TIMEOUT + BUFFER);
    assertEquals("dialog2 never got cursor", 0,
                 dialog2.getCursorSetCount());
    assertEquals("dialog2 never reset cursor", 0,
                 dialog2.getCursorResetCount());
    assertEquals("dialog never got another cursor", 2,
                 dialog.getCursorSetCount());
    assertEquals("dialog never reset another cursor", 2,
                 dialog.getCursorResetCount());
    assertEquals("frame never got cursor", 0,
                 frame.getCursorSetCount());
    assertEquals("frame never reset cursor", 0,
                 frame.getCursorResetCount());
  }

  private class CursorReportingDialog extends Dialog {
    private int cursorResetCount;
    private int cursorSetCount;

    public CursorReportingDialog(Frame owner) {
      super(owner, true);
      init();
    }
    public CursorReportingDialog(Dialog owner) {
      super(owner, "", true);
      init();
    }

    private void init() {
      setCursor(BASE_CURSOR);
      this.cursorSetCount = 0;
      this.cursorResetCount = 0;
    }
    public int getCursorSetCount() {
      return cursorSetCount;
    }
    public int getCursorResetCount() {
      return cursorResetCount;
    }
    public void setCursor(Cursor cursor) {
      super.setCursor(cursor);
      if (BASE_CURSOR.equals(cursor)) {
        cursorResetCount++;
      } else {
        cursorSetCount++;
      }
    }
  }

  private class CursorReportingFrame extends Frame {
    private int cursorResetCount;
    private int cursorSetCount;

    public CursorReportingFrame() {
      super();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    setCursor(BASE_CURSOR);
    cursorSetCount = 0;
    cursorResetCount = 0;
  }

  public int getCursorSetCount() {
    return cursorSetCount;
  }
  public int getCursorResetCount() {
    return cursorResetCount;
  }
  public void setCursor(Cursor cursor) {
    super.setCursor(cursor);
    if (BASE_CURSOR.equals(cursor)) {
      cursorResetCount++;
    } else {
      cursorSetCount++;
    }
  }
}

private abstract class TestRunnable implements Runnable {
  protected boolean testsPassed = true;

  public boolean getTestsPassed() {
    return testsPassed;
  }
}

private class DelayEvent implements Runnable {
  private Runnable callback;
  private int delay;

  public DelayEvent(int delay) {
    this(delay, null);
  }
  public DelayEvent(int delay, Runnable callback) {
    this.delay = delay;
    this.callback = callback;
  }

  public void run() {
    try {
      Thread.sleep(delay);
    } catch (InterruptedException e) {
      e.printStackTrace();
    }
    if (callback != null) {
      callback.run();
    }
  }
}

private class DialogShowEvent implements Runnable {
  private Dialog dialog;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    private boolean visible;
    private int delay;

    public DialogShowEvent(Dialog dialog, boolean visible,
                           int delay) {
      this.dialog = dialog;
      this.visible = visible;
      this.delay = delay;
    }

    public void run() {
      dialog.setVisible(visible);
      if (delay > 0) {
        hangOut(delay);
      }
    }
  }

  private class TestWaitCursorEventQueue
      extends WaitCursorEventQueue {
    private int getDelay;

    public TestWaitCursorEventQueue(int delay) {
      super(delay);
    }

    public AWTEvent getNextEvent() throws InterruptedException {
      if (getDelay > 0) {
        hangOut(getDelay);
        getDelay = 0;
      }
      return super.getNextEvent();
    }
    public void setGetDelay(int getDelay) {
      this.getDelay = getDelay;
    }
  }
}
```

Two types of events are posted to the queue - DelayEvents and DialogShowEvents. DelayEvents are simply events that take some period of time to complete. They can also be configured to call a Runnable callback, to trigger other evaluations or other events. DialogShowEvents either show or hide a dialog, and can also take some period of time after the dialog is shown/hidden to complete. Instances of these two events are passed to postEvent(), along with an event source, and are wrapped in an InvocationEvent and posted to the event queue. Different combinations of these events make up most of the differences between the individual tests.

There are also two GUI classes - CursorReportingDialog and CursorReportingFrame. They will both track cursor sets and resets, and report them to tests that ask. They are usually found in assert() statements.

Finally, you should check out setUp(), tearDown(), flushQueue(), and hangOut(). Past those, everything else is just a test method.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Performance

When this code was first written, I did nothing to examine its performance, so I did not really know how efficient it was. However, that is telling - I have never noticed (as a user) any slowdown because of it. Nevertheless, I have received a question about it (thanks Herman!), which made me realize that I had not examined it. Therefore, I took some time and wrote a unit test that shows the performance numbers for this code. I wrote it as a unit test, just to make it easy to run and in case I ever wanted to really codify what "good performance" meant for this code.

```
import java.awt.*;
import java.awt.event.*;
import java.util.Random;
import javax.swing.*;
import junit.framework.TestCase;

public class WaitCursorEventQueuePerformanceTest
    extends TestCase {
  private static final long FAST = 5;
  private static final long SLOW = 50;
  private static final long MIXED = -1;
  private static final long TIMEOUT = 15;
  private Dialog dialog;
  private Frame frame;

  public WaitCursorEventQueuePerformanceTest(String name) {
   super(name);
  }

  protected void setUp() throws Exception {
   frame = new Frame();
   frame.pack();
   frame.setBounds(-1000, -1000, 100, 100);
   frame.setVisible(true);
   dialog = new Dialog(frame, true);
   dialog.pack();
   dialog.setBounds(-1000, -1000, 100, 100);
  }
  protected void tearDown() throws Exception {
   frame.dispose();
   dialog.dispose();
  }
  private long postEvents(long time) throws InterruptedException {
   InvocationEvent repeatEvent = new InvocationEvent(
       frame, new TimedEvent(time));
   InvocationEvent finalEvent = new InvocationEvent(
       frame, new TimedEvent(time), this, false);
   EventQueue q = Toolkit.getDefaultToolkit().getSystemEventQueue();
   long startTime = System.currentTimeMillis();
   for (int i = 0; i < 1000; i++)
    q.postEvent(repeatEvent);
   synchronized (this) {
    q.postEvent(finalEvent);
    wait(); //we will be notified by finalEvent when it gets posted
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
    long endTime = System.currentTimeMillis();
    return (endTime - startTime);
  }
  public void testNormalPerformanceWithFastEvents()
    throws InterruptedException {
    System.out.println("\nnormal with fast: " + postEvents(FAST));
  }
  public void testNormalPerformanceWithSlowEvents()
    throws InterruptedException {
    System.out.println("\nnormal with slow: " + postEvents(SLOW));
  }
  public void testNormalPerformanceWithMixedEvents()
    throws InterruptedException {
    System.out.println("\nnormal with random: " + postEvents(MIXED));
  }
  public void testWaitQueuePerformanceWithFastEvents()
    throws InterruptedException {
    WaitCursorEventQueue waitQueue = new WaitCursorEventQueue(
                                (int) TIMEOUT);
    Toolkit.getDefaultToolkit().getSystemEventQueue()
      .push(waitQueue);
    System.out.println("\nwait with fast: " + postEvents(FAST));
    waitQueue.close();
  }
  public void testWaitQueuePerformanceWithSlowEvents()
    throws InterruptedException {
    WaitCursorEventQueue waitQueue = new WaitCursorEventQueue(
                                (int) TIMEOUT);
    Toolkit.getDefaultToolkit().getSystemEventQueue()
      .push(waitQueue);
    System.out.println("\nwait with slow: " + postEvents(SLOW));
    waitQueue.close();
  }
  public void testWaitQueuePerformanceWithMixedEvents()
     throws InterruptedException {
    WaitCursorEventQueue waitQueue = new WaitCursorEventQueue(
                                (int) TIMEOUT);
    Toolkit.getDefaultToolkit().getSystemEventQueue()
      .push(waitQueue);
    System.out.println("\nwait with random: " + postEvents(MIXED));
    waitQueue.close();
  }
  public void testWaitQueuePerformanceWithDialogWithFastEvents()
    throws InterruptedException {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        dialog.setVisible(true);
      }
    });
    WaitCursorEventQueue waitQueue = new WaitCursorEventQueue(
                                (int) TIMEOUT);
    Toolkit.getDefaultToolkit().getSystemEventQueue()
      .push(waitQueue);
    System.out.println("\nwait with dialog with fast: " +
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
                postEvents(FAST));
    waitQueue.close();
  }
  public void testWaitQueuePerformanceWithDialogWithSlowEvents()
    throws InterruptedException {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        dialog.setVisible(true);
      }
    });
    WaitCursorEventQueue waitQueue = new WaitCursorEventQueue(
                                (int) TIMEOUT);
    Toolkit.getDefaultToolkit().getSystemEventQueue()
      .push(waitQueue);
    System.out.println("\nwait with dialog with slow: " +
                postEvents(SLOW));
    waitQueue.close();
  }
  public void testWaitQueuePerformanceWithDialogWithMixedEvents()
    throws InterruptedException {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        dialog.setVisible(true);
      }
    });
    WaitCursorEventQueue waitQueue = new WaitCursorEventQueue(
                                (int) TIMEOUT);
    Toolkit.getDefaultToolkit().getSystemEventQueue()
      .push(waitQueue);
    System.out.println("\nwait with dialog with random: " +
                postEvents(MIXED));
    waitQueue.close();
  }

  private class TimedEvent implements Runnable {
    private Random random;
    private long time;

    public TimedEvent(long time) {
      this.time = time;
      if (time == MIXED) {
        random = new Random(1000);
      }
    }

    public void run() {
      try {
        if (time == MIXED) {
          Thread.sleep(
              (long) (random.nextDouble() * (SLOW - FAST) +
                FAST));
        } else {
          Thread.sleep(time);
        }
      } catch (InterruptedException e) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      e.printStackTrace();
    }
   }
  }
}
```

I will leave the analysis of this code up to you, and I will admit that it is not a perfect performance test, but it is good enough for these purposes. You will notice that there are four basic tunable parameters - how long a fast event takes, how long a slow event takes, how long the timer on the WaitCursorEventQueue is, and how many events are used for each test. With the numbers shown here, on my machine, after a reboot, with no other activity, I get:

```
.
normal with fast: 5118
.
normal with slow: 50772
.
normal with random: 27629
.
wait with fast: 5168
.
wait with slow: 50132
.
wait with random: 27660
.
wait with dialog with fast: 5127
.
wait with dialog with slow: 50173
.
wait with dialog with random: 27720

Time: 251.963

OK (9 tests)
```

There are three sets of tests - the normal EventQueue, the WaitCursorEventQueue, and the WaitCursorEventQueue with a modal dialog up (because of the extra processing done while modal dialogs are up). Each set checks the time for 1000 fast events, 1000 slow events, and a series of 1000 random-length events. (Note that because the Random object is seeded with the same number every time, all three random sequences will be the same.)

These results are not significant. Running the test repeatedly gives small variations in the numbers, but there is not any repeatable slowdown demonstratable with the WaitCursorEventQueue. Happily, this confirms my subjective experiences.

Conclusion

This code was hard work, and I did not do it all myself. I did it for my company (who has given me permission to publish it), and I had help. I would like to thank Jason Trump for brainstorming, debugging, and sorting out threading logic. I would also like to thank Ben Schroeder for being so instrumental in teaching me threading so that I could understand it well, years ago.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Well, that's it! You now have an automatic wait cursor manager, which you can install and use in any java program you like. You should not have to think about the wait cursor ever again! I hope this article was clear enough to make it understandable, and not too long. If you have questions, please feel free to email me at truist-waitcursor@truist.com.

Enjoy!

Nathan

Well, I certainly enjoyed editing this newsletter :-) Thanks Nathan, that was a really useful newsletter...

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 076] - Asserting Locks

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 76th edition of **The Java(tm) Specialists' Newsletter**. This week's will be a lot shorter than our mega-newsletter of last week, to give you a break from having to read too much :-)

From the 20th to the 28th of September 2003, I will hopefully be visiting the Tsinghua University in Beijing, China. One of our newsletter readers was invited to present a course on Object Orientation and Java at Tsinghua University to about 50 lecturers and students from all over China. Unfortunately, due to circumstances, he was not able to go to China, and proposed me as a remote proxy instead. Therefore, if you live in Beijing, it would be great to meet with you. Please reply to this email so

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

we can set up a time that will suit everyone. Wo Zhongwen shuo de bu hao. Huitou jian.

Asserting Locks

If my memory serves me correctly, JDK 1.4 was released about 18 months ago. One of the new fangled constructs that JDK 1.4 introduced was the **assert** keyword. In my journeys to companies, I am yet to find one that is actually using Java's **assert** keyword to its full potential.

Here are some reasons why I think this is:

Class Version Changes: To accommodate the assert keyword the class format had to be slightly changed. This means that code containing the assert keyword will only work on JDK 1.4 VMs. Even though this is only a small change, it does mean that old tools may not work with the new classes.

Stability of JDK 1.4: Perhaps companies are reluctant to move over to JDK 1.4.x completely. I know of companies that are still using JDK 1.1 quite happily. They usually adopt the position of "Let's wait and see...", and hope that the next version will have less bugs.

Ignorance: Programmers take a while to change their old habits. I still see code written in 2003 using `java.util.Enumeration` and `java.util.Vector`, usually by experienced Java programmers.

There are obviously more reasons, I would be interested to hear yours.

The reason I hesitate to use assertions is because I have a slight reservation about what will happen to the program when an AssertionError suddenly happens. I have seen code that swallowed Exception, and I would not like developers to start catching Throwable to cater for assertion errors in libraries they use.

However, I do like the ability to switch off assertion checking, even though Pragmatic Programmer advises to keep them switched on.

Let us look at an example of how we could use assertions by starting with a MIDI player example. I received the Sun Core Java Technologies newsletter today, where they showed how to use MIDI from within Java. This is a follow-on from their examples, but heavily refactored.

```
import javax.sound.midi.*;

/**
 * This class represents an Old Song that we might want to record
 * for our listening pleasure.
 */
public abstract class OldSong {
  private final Sequencer sequencer;
  private final Track track;
  private final int resolution;
  private int pos;

  /**
   * @param key is the note that this starts with.  60 is middle C.
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
   * @param tempo is measured in beats per second
   */
  public OldSong(int key, int tempo, int resolution)
        throws MidiUnavailableException,
            InvalidMidiDataException {
    this.resolution = resolution;
    Sequence sequence = new Sequence(Sequence.PPQ, resolution);
    track = sequence.createTrack();
    makeSong(key);
    sequencer = MidiSystem.getSequencer();
    sequencer.open();
    sequencer.setSequence(sequence);
    sequencer.setTempoInBPM(tempo);
  }

  public void start() {
    sequencer.start();
  }

  protected abstract void makeSong(int key)
                    throws InvalidMidiDataException;

  protected void add(int note) throws InvalidMidiDataException {
    add(note, 1);
  }

  protected synchronized void add(int note, int length)
                    throws InvalidMidiDataException {
    addStartEvent(note);
    pos += length;
    addStopEvent(note);
  }

  protected synchronized void addSilence(int length) {
    pos += length;
  }

  /**
   * A piano teacher once told me that the first note in a bar
   * should be emphasized.
   *
   * We assume that resolution has already been set and that we
   * have the "this" monitor.
   */
  private int volume() {
    return ((pos % resolution) == 0) ? 100 : 70;
  }

  /**
   * We assume that we are holding the "this" monitor
   */
  private void addStartEvent(int note)
                throws InvalidMidiDataException {
    ShortMessage message = new ShortMessage();
    message.setMessage(ShortMessage.NOTE_ON, 0, note, volume());
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    track.add(new MidiEvent(message, pos));
  }

  /**
   * We assume that we are holding the "this" monitor
   */
  private void addStopEvent(int note)
                  throws InvalidMidiDataException {
    ShortMessage message = new ShortMessage();
    message.setMessage(ShortMessage.NOTE_OFF, 0, note, 0);
    track.add(new MidiEvent(message, pos));
  }
}
```

You can use that by overriding the class and implementing the makeSong() method to produce your own favourite song. A little bit of musical know-how does help here. In this case, I have put together an approximation of the song "As Time Goes By" from the movie Casablanca. My adaptation goes like this: "I'm not interested in politics - the problems of this world are not in my department. I'm a programmer." If you don't catch that, I suggest you watch the movie :-)

```
import javax.sound.midi.*;

/**
 * This program plays the first few notes of "As Time Goes By"
 * from the movie Casablanca.
 */
public class AsTimeGoesBy extends OldSong {
  public AsTimeGoesBy() throws MidiUnavailableException,
                               InvalidMidiDataException {
    super(65, 20, 8);
  }

  public void makeSong(int key) throws InvalidMidiDataException {
    addSilence(7);
    add(key);
    add(key + 1);
    add(key);
    add(key - 2);
    add(key - 4);
    add(key - 2, 3);
    add(key);
    add(key + 3);
    add(key + 1);
    add(key);
    add(key - 2);
    add(key + 1, 3);
    add(key + 3);
    add(key + 8);
    add(key + 7);
    add(key + 5);
    add(key + 3);
    add(key + 5, 4);
    add(key + 2, 4);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    add(key + 3, 2);
    addSilence(1);
    add(key + 7);
    add(key + 10);
    add(key + 8);
    add(key + 7);
    add(key + 5);
    add(key + 7, 2);
    add(key + 8, 2);
    add(key + 3, 2);
    add(key + 3, 2);
    add(key - 4, 2);
    add(key - 2, 2);
    add(key - 4, 2);
  }
  public static void main(String[] args)
              throws MidiUnavailableException,
                   InvalidMidiDataException {
    OldSong asTime = new AsTimeGoesBy();
    asTime.start();
  }
}
```

In the private methods we have made the assumption that we have the locks for **this**. You can see that we have not synchronized the private methods due to this assumption. Someone editing the class would have to be very careful to read the comment and to understand what a "monitor" is. See my newsletter on "Why I don't read your code comments ..." to see why I think having assertions in comments is dangerous.

However, how can be **assert** that we have the lock inside the method? Thanks to Joshua Bloch, who told Bruce Eckel, who told me, we now have a Thread.hasLock() method. *According to the writer of holdsLock(), it is only intended to be used for assertions, and is not there for performance improvements.*

```
  /**
   * A piano teacher once told me that the first note in a bar
   * should be emphasized.
   */
  private int volume() {
    assert Thread.holdsLock(this);
    assert pos != 0;
    return ((pos % resolution) == 0) ? 100 : 70;
  }

  private void addStartEvent(int note)
                  throws InvalidMidiDataException {
    assert Thread.holdsLock(this);
    ShortMessage message = new ShortMessage();
    message.setMessage(ShortMessage.NOTE_ON, 0, note, volume());
    track.add(new MidiEvent(message, pos));
  }

  private void addStopEvent(int note)
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
                throws InvalidMidiDataException {
    assert Thread.holdsLock(this);
    ShortMessage message =  new ShortMessage();
    message.setMessage(ShortMessage.NOTE_OFF, 0, note, 0);
    track.add(new MidiEvent(message, pos));
  }
```

We now have to compile the files with the -source 1.4 tag, like so:

```
javac -source 1.4 *.java
```

The assertions will be disabled by default. To enable them, we use the -ea switch, like so:

```
java -ea AsTimeGoesBy
```

Non-reentrance

A lot of deadlocks occur because the code is reentrant, meaning that the thread acquires the same lock twice. You could use assertions to check that you have not got a lock before entering a synchronized block, like this:

```
public class A {
  // ...
  public void f() {
    assert !Thread.holdsLock(this);
    synchronized(this) {
      // ...
    }
  }
}
```

Thanks to Bruce Eckel for telling me about Thread.holdsLock().

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 077] - "Wonderfully disgusting hack"

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 77th edition of **The Java(tm) Specialists' Newsletter**. I sent out a newsletter last week to see who preferred text over HTML, especially regarding SPAM filters. Due to the public outcry at the suggestion of moving away from HTML, I will in future send all newsletters as HTML. Please change your SPAM filters to receive emails from my email addresses heinz@javaspecialists.co.za and h.kabutz@computer.org.

I am sitting in a hotel room in a stunning little town in Switzerland called Zug whilst my daughter is celebrating her second birthday today. My voyage began on Saturday afternoon from Cape Town to Johannesburg, and on towards Zurich. My suitcase had other ideas. He liked Johannesburg quite a lot, and together with the other suitcases from the flight, decided to stay overnight and then instead travel to Paris, go on a tour of the Eiffel tower, meet some pretty lady suitcases, etc. He arrived last evening at my hotel door, looking rather bedraggled, with an accusing look on his face: "Why didn't you call?" he told me as he walked through the door. He amused himself by the look of his owner squashed into a one-size-too-small shirt that belonged to our Swiss customer. "What's with the shirt?!" he had the audacity to exclaim. If he thinks he is coming with to China, he has another thing coming!

We are running a Design Patterns Course in Stellenbosch (close to Cape Town) from the 15th to the 17th of this month. Please contact me by return email if you would like more information.

Would your company like to train **lots** of developers in Design Patterns but you do not have the budget to send them on our course? Have you considered licensing our course for internal use? Please speak to us so that we can help you achieve better communication within your development team and improve the skill of your programmers.

"Wonderfully disgusting hack"

People sometimes ask me how I find out the things that I write about in my newsletter. The answer is that I don't know how. It is pretty much a random walk through the JDK source code that gets me the best results.

Last night, whilst my suitcase was screaming through the air at 30'000 feet, I spent some time reading through a new release of the J2EE Core Patterns book. Whilst I was reading it, I was getting more and more agitated with the way that exceptions were being handled. You know I am not a fan of checked exceptions, and in a future newsletter will expound on this. I got sidetracked from reading the book, and ended up reading up on how java.awt handles exceptions. During these explorations, I

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

stumbled across a comment in the java.awt.Toolkit class in JDK 1.4.2 that is simply priceless. It goes like so:

```
static boolean enabledOnToolkit(long eventMask) {
  // Wonderfully disgusting hack for Solaris 9
```

Is that not cute? To write that in code that you *know* could possibly be read by 3'000'000 Java developers around the globe takes a lot of chutzpah.

I did a search on the classes in the JDK 1.4.2 and found 45 classes that contained the word "hack". A great portion of these are from org.apache.* packages, but the majority is from code by Sun Microsystems.

Now the magic question: Am I worried? I am trying to build software for major industry players on top of a basic platform that contains 45 classes with *documented hacks*! Perhaps I should be worried - but I am not. What is worse than knowing you are an idiot? Worse is thinking you are not a dolt, but actually, being one.

Therefore, here are some reasons why I feel more comfortable than ever to write industry-strength code on top of Sun's JDK:

All software contains hacks.
I would prefer to know about them.
Only a **real** developer would write "hack" into his comments.
I am therefore more comfortable using Sun's JDK than some black-box proprietary solution with hundreds of undocumented hacks.

This is a really short newsletter, just to get this topic off my chest. Do yourself a favour and look at the source code of java.awt.Toolkit in JDK 1.4.2. I think you will enjoy it :-)

Warm regards from Switzerland, and my suitcase also says "hi".

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 078] - com.maxoft.memory.MemoryCounter for Java 1.4

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 78th edition of **The Java(tm) Specialists' Newsletter**. Approximately two years ago, I published a newsletter (#29) that showed how we could count the bytes being used by any Java object. I have now turned that newsletter into a little program, that *you* can use to automatically determine memory usage.

This past week I taught UML and Java at the Tsinghua University in Beijing, China. My audience were professors, lecturers and researchers from all around China. They very attentive and participated well during the discussion times. It was a great pleasure spending six days teaching them. After climbing The Great Wall yesterday and meeting the Computer Science Department this morning, I now have some time to churn out yet another newsletter :-)

China is, well, memorable. You simply cannot put it in words. In the week that I have been here, I have already taken about 350 pictures. It is definitely a place that one should visit one day. When I asked one professor how many software developers there were in China, he said "One Million or more". I thought I had misunderstood, so I asked him to repeat. That is a HUGE workforce.

Maximum Solutions (my company) is hoping to do some work together with the Tsinghua University, so maybe you will see some newsletters translated into Mandarin :-)

Advert Alert: Would your company like to train **lots** of developers in Design Patterns but you do not have the budget to send them on our course? Have you considered licensing our course for internal use? We will train your staff on how to train our course. Please speak to us so that we can help you achieve better communication within your development team and improve the skill of your programmers.

com.maxoft.memory.MemoryCounter for Java 1.4

Before I start, I must warn you that I am using classes that will only work for Java 1.4 onwards. In addition, the memory sizes have changed in JDK 1.4, so there are some differences between 1.4 and 1.3. This class is just supposed to give you *some* idea of how big an object is.

After my newsletter #29, I received a few emails telling me that it would be much easier to serialize the object and then count the bytes. If I need to know how much bandwidth my objects will use, then such an approach would make sense. My approach is used for measuring RAM memory. Another advantage of my approach is that the objects do not have to be serializable.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The first class we have is called com.maxoft.memory.MemorySizes and will tell my memory counter class how much memory each element takes. I have determined these through experiments.

```
import java.util.*;

public class com.maxoft.memory.MemorySizes {
  private final Map primitiveSizes =  new IdentityHashMap() {
    {
     put(boolean.class, new Integer(1));
     put(byte.class, new Integer(1));
     put(char.class, new Integer(2));
     put(short.class, new Integer(2));
     put(int.class, new Integer(4));
     put(float.class, new Integer(4));
     put(double.class, new Integer(8));
     put(long.class, new Integer(8));
    }
  };
  public int getPrimitiveFieldSize(Class clazz) {
    return ((Integer) primitiveSizes.get(clazz)).intValue();
  }
  public int getPrimitiveArrayElementSize(Class clazz) {
    return getPrimitiveFieldSize(clazz);
  }
  public int getPointerSize() {
    return 4;
  }
  public int getClassSize() {
    return 8;
  }
}
```

Next we have the class that counts the memory sizes. For an explanation of this code, I would suggest you look at the original newsletter.

```
import java.lang.reflect.*;
import java.util.*;

/**
 * This class can estimate how much memory an Object uses.  It is
 * fairly accurate for JDK 1.4.2.  It is based on the newsletter #29.
 */
public final class com.maxoft.memory.MemoryCounter {
  private static final com.maxoft.memory.MemorySizes sizes =  new
com.maxoft.memory.MemorySizes();
  private final Map visited =  new IdentityHashMap();
  private final Stack stack =  new Stack();

  public synchronized long estimate(Object obj) {
    assert visited.isEmpty();
    assert stack.isEmpty();
    long result = _estimate(obj);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    while (!stack.isEmpty()) {
      result += _estimate(stack.pop());
    }
    visited.clear();
    return result;
  }

  private boolean skipObject(Object obj) {
    if (obj instanceof String) {
      // this will not cause a memory leak since
      // unused interned Strings will be thrown away
      if (obj == ((String) obj).intern()) {
        return true;
      }
    }
    return (obj == null)
        || visited.containsKey(obj);
  }

  private long _estimate(Object obj) {
    if (skipObject(obj)) return 0;
    visited.put(obj, null);
    long result = 0;
    Class clazz = obj.getClass();
    if (clazz.isArray()) {
      return _estimateArray(obj);
    }
    while (clazz != null) {
      Field[] fields = clazz.getDeclaredFields();
      for (int i = 0; i < fields.length; i++) {
        if (!Modifier.isStatic(fields[i].getModifiers())) {
          if (fields[i].getType().isPrimitive()) {
            result += sizes.getPrimitiveFieldSize(
                fields[i].getType());
          } else {
            result += sizes.getPointerSize();
            fields[i].setAccessible(true);
            try {
              Object toBeDone = fields[i].get(obj);
              if (toBeDone != null) {
                stack.add(toBeDone);
              }
            } catch (IllegalAccessException ex) { assert false; }
          }
        }
      }
      clazz = clazz.getSuperclass();
    }
    result += sizes.getClassSize();
    return roundUpToNearestEightBytes(result);
  }

  private long roundUpToNearestEightBytes(long result) {
    if ((result % 8) != 0) {
      result += 8 - (result % 8);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
    return result;
  }

  protected long _estimateArray(Object obj) {
    long result = 16;
    int length = Array.getLength(obj);
    if (length != 0) {
      Class arrayElementClazz = obj.getClass().getComponentType();
      if (arrayElementClazz.isPrimitive()) {
        result += length *
            sizes.getPrimitiveArrayElementSize(arrayElementClazz);
      } else {
        for (int i = 0; i < length; i++) {
          result += sizes.getPointerSize() +
              _estimate(Array.get(obj, i));
        }
      }
    }
    return result;
  }
}
```

I recently changed this code significantly to include the new IdentityHashMap and to add support for interned Strings. This would not have been possible without a good suite of unit tests. I normally do not include my unit tests in the newsletters, since they do use a lot of space. However, I would like to **hear from you** if you have a JDK 1.4.x where the results are different. The unit tests pass on my machine, using JDK 1.4.2, build 1.4.2-b28.

```
import java.util.*;
import junit.framework.TestCase;
import junit.swingui.TestRunner;

public class com.maxoft.memory.MemoryCounterTest extends TestCase {
  public static void main(String[] args) {
    TestRunner.run(com.maxoft.memory.MemoryCounterTest.class);
  }

  private static final com.maxoft.memory.MemoryCounter mc = new
com.maxoft.memory.MemoryCounter();

  public com.maxoft.memory.MemoryCounterTest(String name) {
    super(name);
  }

  public void testString() {
    assertEquals(64, mc.estimate(new String("Hello World!")));
  }

  public void testIntegerToString() {
    for (int i=0; i<1; i++) {
      assertEquals(72, mc.estimate("" + i));
    }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }

  static class Entry implements Map.Entry {
    final Object key;
    Object value;
    final int hash;
    Entry next;

    Entry(int h, Object k, Object v, Entry n) {
      value = v;
      next = n;
      key = k;
      hash = h;
    }

    public Object getKey() {
      return key;
    }

    public Object getValue() {
      return value;
    }

    public Object setValue(Object value) {
      return value;
    }
  }

  public void testHashMap() {
    assertEquals(120, mc.estimate(new HashMap()));

    Byte[] all = new Byte[256];
    for (int i = -128; i < 128; i++) {
      all[i + 128] = new Byte((byte) i);
    }
    assertEquals(5136, mc.estimate(all));

    HashMap hm = new HashMap();
    for (int i = -128; i < 128; i++) {
      hm.put("" + i, new Byte((byte) i));
    }
    assertEquals(30776, mc.estimate(hm));
  }

  public void testVector() {
    assertEquals(80, mc.estimate(new Vector(10)));
  }

  public void testObject() {
    assertEquals(8, mc.estimate(new Object()));
  }

  public void testInteger() {
    assertEquals(16, mc.estimate(new Integer(1)));
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public void testCharArray() {
  assertEquals(40, mc.estimate("Hello World!".toCharArray()));
}

public void testByte() {
  assertEquals(16, mc.estimate(new Byte((byte) 10)));
}

public void testThreeBytes() {
  assertEquals(16, mc.estimate(new ThreeBytes()));
}

public void testSixtyFourBooleans() {
  assertEquals(72, mc.estimate(new SixtyFourBooleans()));
}

public void testThousandBooleansObjects() {
  Boolean[] booleans = new Boolean[1000];

  for (int i = 0; i < booleans.length; i++)
    booleans[i] = new Boolean(true);

  assertEquals(20016, mc.estimate(booleans));
}

public void testThousandBytes() {
  assertEquals(1016, mc.estimate(new byte[1000]));
}

public void testEmptyArrayList() {
  assertEquals(80, mc.estimate(new ArrayList()));
}

public void testFullArrayList() {
  ArrayList arrayList = new ArrayList(10000);

  for (int i = 0; i < 10000; i++) {
    arrayList.add(new Object());
  }

  assertEquals(120040, mc.estimate(arrayList));
}

public void testFullLinkedList() {
  LinkedList linkedList = new LinkedList();

  for (int i = 0; i < 10000; i++) {
    linkedList.add(new Object());
  }

  assertEquals(320048, mc.estimate(linkedList));
}

public void testComplexClass() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    assertEquals(48, mc.estimate(new ComplexClass()));
  }

  public void testBooleanArray() {
    assertEquals(27, mc.estimate(new boolean[11]));
  }

  public void testShortArray() {
    assertEquals(38, mc.estimate(new short[11]));
  }

  public void testIntArray() {
    assertEquals(60, mc.estimate(new int[11]));
  }

  public void testFloatArray() {
    assertEquals(60, mc.estimate(new float[11]));
  }

  public void testLongArray() {
    assertEquals(104, mc.estimate(new long[11]));
  }

  public void testDoubleArray() {
    assertEquals(104, mc.estimate(new double[11]));
  }

  static class ThreeBytes {
    byte b0;
    byte b1;
    byte b2;
  }

  private static class ComplexClass {
    ComplexClass cc = this;
    boolean z;
    byte b;
    char c;
    double d;
    float f;
    int i;
    long l;
    short s;
  }

  private static class SixtyFourBooleans {
    boolean a0;
    boolean a1;
    boolean a2;
    boolean a3;
    boolean a4;
    boolean a5;
    boolean a6;
    boolean a7;
    boolean b0;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
boolean b1;
boolean b2;
boolean b3;
boolean b4;
boolean b5;
boolean b6;
boolean b7;
boolean c0;
boolean c1;
boolean c2;
boolean c3;
boolean c4;
boolean c5;
boolean c6;
boolean c7;
boolean d0;
boolean d1;
boolean d2;
boolean d3;
boolean d4;
boolean d5;
boolean d6;
boolean d7;
boolean e0;
boolean e1;
boolean e2;
boolean e3;
boolean e4;
boolean e5;
boolean e6;
boolean e7;
boolean f0;
boolean f1;
boolean f2;
boolean f3;
boolean f4;
boolean f5;
boolean f6;
boolean f7;
boolean g0;
boolean g1;
boolean g2;
boolean g3;
boolean g4;
boolean g5;
boolean g6;
boolean g7;
boolean h0;
boolean h1;
boolean h2;
boolean h3;
boolean h4;
boolean h5;
boolean h6;
boolean h7;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
  }
}
```

I would be happy to hear from you if this newsletter is useful to you. Just pop me a quick email telling me how this can help you.

Kind regards

Heinz

P.S. My original code has loads of comments. I stripped them out for the newsletter to save space.

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 079] - Generic toString()

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 79th edition of **The Java(tm) Specialists' Newsletter**. My expedition to China was an adventure. At the Beijing Airport, on my way home, I was told that my seat from Singapore to Cape Town had been cancelled due to some technicality. I really did not look forward to being stranded in Singapore for one week, so some serious prayers went up in China and Cape Town. Within one hour, my excellent travel agent managed to get the seat reinstated and confirmed.

Sitting outside on my balcony today, overlooking False Bay (the worlds favourite breeding ground for the Great White Shark), I want to give some free advice if you ever want to purchase a house in Cape Town. We have a strong south-easterly wind that blows most of summer. Looking out over False Bay, I can see "white horses" on the water, which tells me that the wind is fairly strong today. Where we live there is just a gentle breeze. You should look for areas where the wind is not too strong so that you can work outside with your notebook during summer :-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Advert Alert: Would your company like to encourage your developers to use Design Patterns but you do not have the budget to send them all on a course? Have you considered licensing our course for internal use? We would love to hear from you if you would like to achieve better communication within your development team and improve the skills of your programmers.

Generic toString()

I would like to thank Ravi Nukala for the idea in this newsletter.

This newsletter explores a way in which you can use a generic toString() method that takes any Java object and converts it to a String object, or appends it to a StringBuffer. A restriction in the current system is that circular handles will result in infinite recursion, which cause stack overflow errors in Java. This could perhaps be addressed by keeping a history of which objects had been visited. It complicates the system substantially and this is why I decided not to cater for this.

When I ask a big audience which Design Patterns they know, I usually get at least the following three responses: Singleton, Factory and Facade. Singleton is very comfortable because it seems object-oriented, yet the way that most people use it is not. Factory does not appear "as is" in the Gang-of-Four (GoF, or Erich Gamma et al) book. There is an Abstract Factory and a Factory Method, both of which are very different to the static creation method that is commonly called *"The Factory"*. Then there is the Facade pattern, which I personally maintain is not really a Design Pattern in the classical sense. If you read the book, you will notice that the structure section of the Facade pattern is markedly different to the other patterns. There is a reason for this. The Facade pattern is there to help you sort out the huge number of classes that you get when you use a lot of patterns. The Facade pattern often confused with an interface to a subsystem. The Facade is more than just an interface. It should also allow you direct access to the subsystem, something that an interface would usually not allow. In the sample code of this newsletter, we will see a Facade, that "makes the subsystem easier to use".

Our Facade constructs three Converters and sets them up in a Chain-Of-Responsibility. It then passes each object to the chain, which deals with it by calling the "handle()" method.

```
/**
 * The default behaviour is to start with the NullConverter,
 * followed by the ArrayConverter and then the ObjectConverter.
 */
public class ToStringFacade {
  private final static ToStringConverter INITIAL =
      new NullConverter(
          new ArrayConverter(
              new ObjectConverter(null)));
  public static String toString(Object o) {
    return toString(o, new StringBuffer()).toString();
  }
  public static StringBuffer toString(Object o, StringBuffer buf) {
    return INITIAL.handle(o, buf);
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

As a start, we define a Utils class that returns the class name without the package. This utilities class is used in the various converters.

```
public class Utils {
  /** We only want to see the class name, without the package. */
  public static String stripPackageName(Class c) {
    return c.getName().replaceAll(".*\\.", "").replace('$', '.');
  }
}
```

Next we define an abstract class that we want to use as the main handler in this Chain-of-Responsibility pattern. This avoids having to have a multi-conditional if-else statement that has to decide which Strategy to use for a particular object.

```
/**
 * Here we use the Chain-of-Responsibility design pattern that is
 * very similar to the Strategy pattern, except that we pass the
 * request on if we cannot deal with it.  It helps us to reduce
 * the number of multi-conditional if-else-if statements that we
 * need in our program to decide which strategy to use.
 */
public abstract class ToStringConverter {
  private final ToStringConverter successor;
  /** We need to know the successor, if this handler cannot cope
   * with the request. */
  protected ToStringConverter(ToStringConverter successor) {
    this.successor = successor;
  }
  /** handle() decides whether this current object can handle the
   * request; otherwise it is passed onto the next in the sequence */
  protected final StringBuffer handle(Object o, StringBuffer buf) {
    if (!isHandler(o)) {
      assert successor != null;
      return successor.handle(o, buf);
    }
    return toString(o, buf);
  }
  /** Subclasses can specify whether they can handle the current
   * object. */
  protected abstract boolean isHandler(Object o);
  /** The toString() method is the main method that is called from
   * within the handle() method, once it is established which object
   * should handle the request. */
  protected StringBuffer toString(Object o, StringBuffer buf) {
    buf.append('(');
    appendName(o, buf);
    buf.append(getSeparator());
    appendValues(o, buf);
    buf.append(')');
    return buf;
  }
  /** The separator between name and value can be different for
   * different types of objects. */
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  protected char getSeparator() {  return '='; }
  /** This method will determine an identifier for the current object. */
  protected void appendName(Object o, StringBuffer buf) { }
  /** This method will determine the values for the current object. */
  protected void appendValues(Object o, StringBuffer buf) { }
}
```

The easiest case to deal with is when the object is **null**. In this case we always let toString() append "(null)":

```
/** This class follows the Null Object Pattern by Bobby Woolf. */
public class NullConverter extends ToStringConverter {
  public NullConverter(ToStringConverter successor) {
    super(successor);
  }
  /** This handler is only used if the object is null */
  protected boolean isHandler(Object o) {
    return o == null;
  }
  protected StringBuffer toString(Object o, StringBuffer buf) {
    buf.append("(null)");
    return buf;
  }
}
```

The next case to deal with is when the object is an array. We would like to be able to deal with multi-dimensional arrays. In addition, we want to ensure that we can deal with classes that themselves have not got their own toString() method. We therefore use our own ToStringConverter class unless the array components are primitives, other arrays or Strings.

```
import java.lang.reflect.Array;

/** This converter only supports Arrays. It supports
 * primitive arrays and multi-dimensional arrays. */
public class ArrayConverter extends ToStringConverter {
  public ArrayConverter(ToStringConverter successor) {
    super(successor);
  }
  /** This handler only works for arrays. */
  protected boolean isHandler(Object o) {
    return o.getClass().isArray();
  }
  /** We want to append the type of the array and the number of
   * dimensions, e.g. for a three-dimensional array we want to
   * append [][][].  Using += for the postfix String is not
   * ideal, but the most common case will probably be a single
   * dimension array, in which case there would not be any
   * concatenation of Strings. */
  protected void appendName(Object o, StringBuffer buf) {
    assert o.getClass().isArray();
    String postfix = "[]";
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    Class c = o.getClass().getComponentType();
    while(c.isArray()) {
      postfix += "[]";
      c = c.getComponentType();
    }
    buf.append(Utils.stripPackageName(c));
    buf.append(postfix);
  }
  /** We show the array using the dimensions and the toString()
   * methods of the values.  This method is recursive to handle multi-
   * dimensional arrays. */
  protected void appendValues(Object o, StringBuffer buf) {
    assert o.getClass().isArray();
    buf.append('{');
    int length = Array.getLength(o);
    for (int i = 0; i < length; i++) {
      Object value = Array.get(o, i);
      if (value != null && value.getClass().isArray()) {
        appendValues(value, buf);
      } else if (o.getClass().getComponentType().isPrimitive()) {
        buf.append(value);
      } else if (value instanceof String) {
        buf.append("").append(value).append("");
      } else {
        ToStringFacade.toString(value, buf);
      }
      if (i < length - 1) {
        buf.append(',');
      }
    }
    buf.append('}');
  }
}
```

The most complicated handler is for general objects. We want to show the class name and all the fields and their values. This is done recursively.

```
import java.lang.reflect.*;
import java.util.*;

public class ObjectConverter extends ToStringConverter {
  public ObjectConverter(ToStringConverter successor) {
    super(successor);
  }
  /** This is the end-point of the chain.  If we get here, we
   * use this handler. */
  protected boolean isHandler(Object o) {
    return true;
  }
  /** We specify a different separator for between name and
   * values to make the output look nicer. */
  protected char getSeparator() { return ':'; }
  /** For the name of the class, we strip off the package name. */
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  protected void appendName(Object o, StringBuffer buf) {
    buf.append(Utils.stripPackageName(o.getClass()));
  }
  /** The values are a bit more complicated.  We first have to
   * find all fields.  To find all private fields, we have to
   * recurse up the hierarchy of the object.  We then go through
   * all the fields and append the name and the value.  Unless
   * we have reached the end, we append ", ". */
  protected void appendValues(Object o, StringBuffer buf) {
    Iterator it = findAllFields(o);
    while(it.hasNext()) {
      Field f = (Field) it.next();
      appendFieldName(f, o, buf);
      buf.append('=');
      appendFieldValue(f, o, buf);
      if (it.hasNext()) {
        buf.append(", ");
      }
    }
  }
  /** If the field's class is not the object's class (i.e. the
   * field is declared in a superclass) then we print out the
   * superclass' name together with the field name. */
  private void appendFieldName(Field f, Object o, StringBuffer buf) {
    if (f.getDeclaringClass() != o.getClass()) {
      buf.append(Utils.stripPackageName(f.getDeclaringClass()));
      buf.append('.');
    }
    buf.append(f.getName());
  }
  /** We set the field to be "accessible", i.e. public.  If the
   * type of the field is primitive, we simply append the value;
   * otherwise we recursively print the value using our
   * ToStringFacade. */
  private void appendFieldValue(Field f, Object o, StringBuffer buf) {
    try {
      f.setAccessible(true);
      Object value = f.get(o);
      if (f.getType().isPrimitive()) {
        buf.append(value);
      } else {
        ToStringFacade.toString(value, buf);
      }
    } catch (IllegalAccessException e) {
      assert false : "We have already set it accessible!";
    }
  }
  /** Find all fields of an object, whether private or public.
   * We also look at the fields in super classes. */
  private Iterator findAllFields(Object o) {
    Collection result = new LinkedList();
    Class c = o.getClass();
    while (c != null) {
      Field[] f = c.getDeclaredFields();
      for (int i = 0; i < f.length; i++) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      if (!Modifier.isStatic(f[i].getModifiers())) {
        result.add(f[i]);
      }
    }
    c = c.getSuperclass();
  }
  return result.iterator();
 }
}
```

The last class that we need is the very necessary unit test. This should enable you to refactor the classes to your heart's content, and not be too worried about breaking anything.

```
import java.util.*;
import junit.framework.TestCase;
import junit.swingui.TestRunner;

public class ToStringTest extends TestCase {
 public static void main(String[] args) {
  TestRunner.run(ToStringTest.class);
 }
 public ToStringTest(String name) {
  super(name);
 }
 public void testPackageStripping() {
  assertEquals("Integer",
     Utils.stripPackageName(Integer.class));
  assertEquals("Map.Entry",
     Utils.stripPackageName(Map.Entry.class));
  assertEquals("ToStringTest",
     Utils.stripPackageName(ToStringTest.class));
 }
 public void testNull() {
  assertEquals("(null)", ToStringFacade.toString(null));
 }
 public void testInteger() {
  assertEquals("(Integer:value=42)",
     ToStringFacade.toString(new Integer(42)));
 }
 public void testString() {
  assertEquals("(String:value=(char[]={H,e,l,l,o, ,W,o,r,l,d,!}), " +
     "offset=0, count=12, hash=0)",
     ToStringFacade.toString("Hello World!"));
 }
 public void testArray() {
  assertEquals("(int[]={1,2,3})",
     ToStringFacade.toString(new int[]{1, 2, 3}));
 }
 public void testMultiArray() {
  assertEquals("(long[][][][]={{{{1,2,3},{4}},{{5}}}})",
     ToStringFacade.toString(
        new long[][][][]{{{{1, 2, 3}, {4}}, {{5}}}}));
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  }
  public void testTestClass() {
    assertEquals("(ToStringTest.TestClass:names=" +
        "(String[]={\"Heinz\",\"Joern\",\"Pieter\",\"Herman\"" +
        ",\"John\"}), totalIQ=900)",
        ToStringFacade.toString(new TestClass()));
  }
  private static class TestClass {
    private final String[] names = {"Heinz", "Joern", "Pieter",
                                    "Herman", "John"};
    private final int totalIQ = 180 * 5;
  }
  public void testArrayList() {
    ArrayList list = new ArrayList();
    list.add("Heinz");
    list.add("Helene");
    list.add("Maxi");
    list.add("Connie");
    assertEquals("(ArrayList:elementData=(Object[]={\"Heinz\"" +
        ",\"Helene\",\"Maxi\",\"Connie\",(null),(null),(null)," +
        "(null),(null),(null)}), size=4, AbstractList.modCount=4)",
        ToStringFacade.toString(list));
  }
}
```

That concludes this week's newsletter. My disclaimer this week is that I have not used this approached in a production system. It is probably quite inefficient to do it this way and there could be ways in which you can optimise it. It is sometimes convenient to have a method that will dump the contents of an object, even private fields of the superclasses.

Next week I will show you how it would be possible to have many public classes in one ".java" file, and I will also attempt to convince you why it is not a good idea to do that :-)

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 080] - Many Public Classes in One File

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 80th edition of **The Java(tm) Specialists' Newsletter**. There is a saying in the English language: It never rains, but it pours. This means that either I am unbelievably busy, or I have time to write several newsletters at once. So, at the moment the sun is shining and I have the luxury of being able to relax with my hobby of writing newsletters.

The idea in this newsletter was prompted by a professor who attended my Java classes in China. I was talking about Java only allowing one public class per file, and he asked: "What about inner classes?" The resulting discussion ended up in this newsletter. Technically, it is possible to have lots of public static inner classes and use them as normal public classes, but I would strongly recommend against that.

Advert Alert: Would your company like to encourage your developers to use Design Patterns but you do not have the budget to send them all on a course? Have you considered licensing our course for internal use? We would love to hear from you if you would like to achieve better communication within your development team and improve the skills of your programmers.

We are getting more organised. We have now got a schedule of when our courses are happening. I have resisted doing this for five years, because often these schedules are closer to fiction than reality. We will try to stick to these dates, but we are flexible enough to move them if you need us to.

Many Public Classes in One File

Some of the most common complaints that I hear about Java are these:

Java is slow.
You cannot make an "exe".
Every Java class has to be in a separate file.

I sometimes struggle to take these complaints seriously. Here are my standard answers:

The speed of execution depends greatly on the skill of the developer. I have heard of real-world applications where the Java application was faster than an equivalent C++ application. For number crunching, you will probably do better with C++, but for general applications, the advantages of Java outweigh the slight performance degradation that you may experience.
There are several compilers available that will produce an "exe" from Java classes. Some of these are fairly powerful. In a future newsletter we are planning to review some of them.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Usually my answer here is that it makes source control easier when you have a separate file for each class. As described in my newsletter "Once Upon an Oak", a reason for this restriction is to make the compilation process faster. This newsletter will demonstrate an approach where you can have several public classes in one .java file.

Before I show you this approach, please understand that I do not endorse writing classes in this way. I prefer using one file per class, even for non-public classes. I once had to work on a class that had about one hundred inner classes. It was a nightmare.

We start by defining package com.maxoft.tjsn which contains class All. This class contains all the classes as **public static** inner classes.

```
package com.maxoft.tjsn;

public class All {
  public static class A {
    public void f() {}
  }
  public static class B {
    public void g() {}
  }
  public static class C {
    public void h() {}
  }
  public static class D extends A {
    public void f() {}
  }
  public static class E {
    public void jump() {}
  }
  public static class F {
    private final E e = new E();
    public void skip() {
     e.jump();
    }
  }
}
```

It is important that the class is in a package, as we will see just now. Next we show the first approach of how we could now use classes A,B,C,etc.

```
import com.maxoft.tjsn.All;

public class AllTest1 {
  public static void main(String[] args) {
    All.A a = new All.A();
    All.B b = new All.B();
    All.C c = new All.C();
    All.D d = new All.D();
    All.E e = new All.E();
    All.F f = new All.F();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
  }
}
```

Admittedly, this is ugly. However, there is another approach. We can import the inner classes directly, like so:

```
import com.maxoft.tjsn.All.*;

public class AllTest2 {
  public static void main(String[] args) {
    A a = new A();
    B b = new B();
    C c = new C();
    D d = new D();
    E e = new E();
    F f = new F();
  }
}
```

This is now a lot prettier than saying All.A and All.B. I think it only works if the class is in a package. This should not be a restriction since classes should be in packages anyway.

But, rather don't do this...

Here are some reasons why you should rather not do this: First, it is going to result in large source files. Second, all members of the classes will effectively only have package access protection. This can easily result in spaghetti code that would make an Italian Mama proud. Third, most IDEs will struggle to support this well. Fourth, I seem to recall that some compilers cannot compile importing of inner classes.

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 081] - Catching Exceptions in GUI Code

Author: Dr. Heinz M. Kabutz

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 81st edition of **The Java(tm) Specialists' Newsletter**. I can hardly believe that it is now already over six weeks since I wrote the last newsletter. Since then, the Grupo de Usu•˜rios Java - GUJ, from Brazil has started translating some of our newsletters into Portuguese. You can see the result on our archive webpage. Please send us an email if you would like to translate the newsletter into your language.

After an extremely busy September lecturing Java and Design Patterns, and a busy October and November programming and fixing bugs, we are again looking at putting together some courses. Please contact me if we should include you in our planning.

Design Patterns are the key to *really* understanding object orientation. What is the difference between the Gang-of-Four Design Patterns and J2EE Patterns? The Gang-of-Four patterns represent the basic building blocks of object orientation. Most of the patterns rely on polymorphism. The J2EE patterns, on the other hand, are design solutions for the J2EE architecture. They are at a far higher level than the Gang-of-Four patterns. Knowing the Gang-of-Four patterns will help you understand how the J2EE patterns work.

Maximum Solutions is not only known for **The Java(tm) Specialists' Newsletter**, but also for presenting the best training on Design Patterns that you will find anywhere :-) Please contact us so that we can put together a proposal for you, with no obligation!

Catching Exceptions in GUI Code

One of the reasons that I continuously promote Design Patterns is that they really work. A pattern that I enjoy teaching is the Command pattern, because the exercise with this pattern is particularly hard unless your mind has already been altered. During our discussion time around that pattern, I usually pose this questions: *"What happens when our command causes an exception?"*

The answer is not that obvious. The invoker usually does not know what to do with an uncaught exception. Unless you have an alternative exception listening mechanism in place, the exception will simply be lost.

Let us take the following GUI as an example:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;

public class Gui extends JFrame {
  private static final int[] DAYS_PER_MONTH =
     {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
  public Gui() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    super("GUI Example");
    final JTextArea text = new JTextArea(14, 30);
    getContentPane().add(new JScrollPane(text));
    getContentPane().add(new JButton(new AbstractAction("Calculate") {
      public void actionPerformed(ActionEvent e) {
        text.setText("");
        for (int i=0; i<=DAYS_PER_MONTH.length; i++) {
          text.append("Month " + (i+1) + ": " +
              DAYS_PER_MONTH[i] + "\n");
        }
      }
    }), BorderLayout.NORTH);
  }
  public static void main(String[] args) {
    Gui gui = new Gui();
    gui.pack();
    gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    gui.show();
  }
}
```

Please compile and run this program using javaw.exe instead of java.exe. i.e. type the following:

```
javaw Gui
```

This will start the Gui frame and wait for you to click the button. Now it gets really interesting! When you press the button, you see the months and days per month appearing on the textpane. It all seems good, except that the button looks like it stays depressed.

Please run the program again using java.exe, and when you press the button you will see the following exception on the console:

```
java.lang.ArrayIndexOutOfBoundsException: 12
  at Gui$1.actionPerformed(Gui.java:16)
  at javax.swing.AbstractButton.fireActionPerformed
  at javax.swing.AbstractButton$ForwardActionEvents.actionPerformed
  at javax.swing.DefaultButtonModel.fireActionPerformed
  at javax.swing.DefaultButtonModel.setPressed
  at javax.swing.plaf.basic.BasicButtonListener$ReleasedAction.actionPerformed
  at javax.swing.SwingUtilities.notifyAction
  at javax.swing.JComponent.processKeyBinding
  at javax.swing.JComponent.processKeyBindings
  at javax.swing.JComponent.processKeyEvent
  at java.awt.Component.processEvent
  at java.awt.Container.processEvent
  at java.awt.Component.dispatchEventImpl
  at java.awt.Container.dispatchEventImpl
  at java.awt.Component.dispatchEvent
  at java.awt.KeyboardFocusManager.redispatchEvent
  at java.awt.DefaultKeyboardFocusManager.dispatchKeyEvent
  at java.awt.DefaultKeyboardFocusManager.preDispatchKeyEvent
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  at java.awt.DefaultKeyboardFocusManager.typeAheadAssertions
  at java.awt.DefaultKeyboardFocusManager.dispatchEvent
  at java.awt.Component.dispatchEventImpl
  at java.awt.Container.dispatchEventImpl
  at java.awt.Window.dispatchEventImpl
  at java.awt.Component.dispatchEvent
  at java.awt.EventQueue.dispatchEvent
  at java.awt.EventDispatchThread.pumpOneEventForHierarchy
  at java.awt.EventDispatchThread.pumpEventsForHierarchy
  at java.awt.EventDispatchThread.pumpEvents
  at java.awt.EventDispatchThread.pumpEvents
  at java.awt.EventDispatchThread.run
```

Besides printing the thread stack trace to the console, the Java event dispatch thread also does another thing: *it dies!* I only discovered that recently. The dispatch thread dying is probably the reason that the button stays depressed. If you look in the EventDispatchThread code, you will see that a new thread is only created once more Gui events occur. Unless you move the mouse, or press a key, no events will happen and so the button stays depressed.

We can verify that the event dispatch thread dies by printing out the System.identityHashCode() of the thread that executes the actionPerformed method (this should be the EventDispatchThread). In this example, the exception occurs after the button is pressed for the fourth time.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;

public class ThreadsDie extends JFrame {
  public ThreadsDie() {
    super("GUI Example");
    final JTextArea text = new JTextArea(14, 30);
    getContentPane().add(new JScrollPane(text));
    getContentPane().add(new JButton(new AbstractAction("Calculate") {
      private int countdown = 3;
      public void actionPerformed(ActionEvent e) {
        text.append("Event Queue Thread: " +
            System.identityHashCode(Thread.currentThread()));
        text.append("\n");
        if (--countdown <= 0) {
          throw new IllegalArgumentException();
        }
      }
    }), BorderLayout.NORTH);
  }
  public static void main(String[] args) {
    ThreadsDie gui = new ThreadsDie();
    gui.pack();
    gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    gui.show();
  }
}
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

The output on the textpane is something like:

```
Event Queue Thread: 29959477
Event Queue Thread: 29959477
Event Queue Thread: 29959477
Event Queue Thread: 23994289
Event Queue Thread: 19764978
Event Queue Thread: 1114115
Event Queue Thread: 1565898
Event Queue Thread: 11383252
Event Queue Thread: 23110255
Event Queue Thread: 21514757
```

Creating a new thread each time you get an exception in your Gui is not that good, since it is fairly expensive to create threads. We would like to get to the position where no exceptions are generated by our Gui at all.

Last Friday I went to a customer who had an old version of the software I am working on at the moment. When it started up, I saw several exceptions appear on the console. On questioning him about that he said: "Oh those, they always appear. The program still works though."

This makes me scared. I would rather get 259 emails in my inbox telling me that something is seriously amiss, than have my customers think that seeing exceptions on the console is acceptable.

However, how can we catch all uncaught exceptions that are generated in the Gui?

The trick lies in a class called the ThreadGroup and in the way that JDK 1.4 now constructs the Gui threads. A Thread may belong to a ThreadGroup. ThreadGroup has a method called uncaughtException(Thread t, Throwable e) that we can override and which is called whenever an uncaught exception occurs. The event dispatch thread still dies, but at least we know about it.

```
import javax.swing.*;
import java.awt.*;

public class ExceptionGroup extends ThreadGroup {
  public ExceptionGroup() {
    super("ExceptionGroup");
  }
  public void uncaughtException(Thread t, Throwable e) {
    JOptionPane.showMessageDialog(findActiveFrame(),
       e.toString(), "Exception Occurred", JOptionPane.OK_OPTION);
    e.printStackTrace();
  }
  /**
   * I hate ownerless dialogs.  With this method, we can find the
   * currently visible frame and attach the dialog to that, instead
   * of always attaching it to null.
   */
  private Frame findActiveFrame() {
    Frame[] frames = JFrame.getFrames();
    for (int i = 0; i < frames.length; i++) {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
      Frame frame = frames[i];
      if (frame.isVisible()) {
        return frame;
      }
    }
    return null;
  }
}
```

All we now need to do is start the Gui from within a Thread that belongs to this ExceptionGroup, and we will catch all uncaught exceptions that are caused by the event dispatch thread:

```
import javax.swing.*;

public class BetterGui {
  public static void main(String[] args) {
    ThreadGroup exceptionThreadGroup = new ExceptionGroup();
    new Thread(exceptionThreadGroup, "Init thread") {
      public void run() {
        Gui gui = new Gui();
        gui.pack();
        gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        gui.show();
      }
    }.start();
  }
}
```

This solution is far from perfect. In my industrial-strength exception framework (not available for general use, don't ask), the uncaught exceptions go to a central place from where they are dispatched to the correct listeners. Those might be more fancy dialogs, or emails sent to the support team, or a stacktrace in a log file.

That is it for today. I need some beauty sleep so that tomorrow I can tackle all the exceptions that we found with this approach in some existing code.

Kind regards

Heinz

*P.S. Please don't forget to contact me if you would like to translate **The Java(tm) Specialists' Newsletter** into your native language*

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 082] - TristateCheckBox based on the Swing JCheckBox

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 82nd edition of **The Java(tm) Specialists' Newsletter**. My last newsletter caused some subscribers to come forward, offering to translate to their language. The most interesting one, in my opinion, is our Zulu translation. Zulu is spoken by approximately 9 million people in Southern Africa. It is one of the 11 official languages in South Africa. I am particularly grateful to **Mondli Mabaso** for sacrificing his time and bringing us the translation.

In addition to Zulu, we have also been approached with Estonian, Polish and Bulgarian. Thank you very much in advance! Please send us an email if you would like to translate the newsletter into your language.

My uncle Karl-Heinz is one of many relatives who is on the subscriber list. Having a large family is an advantage with electronic newsletters, since you immediately have a captive audience, who dare not unsubscribe for fear of mortally offending you *grin*. Karl-Heinz and my aunt Gunhild visited us in November from Germany, and one of the reasons the newsletters have been so scarce is because they had me chase a little white ball across the grass (or rather the bushes) in Somerset West :-)

TristateCheckBox based on the Swing JCheckBox

I wrote this component for a customer a few years ago and it has been running happily in production since then. Thank you for letting me publish this! I have found and removed a bug at the same time :-)

Writing custom components in Swing can be tricky, especially when you are trying to change the behaviour in a Look & Feel independent way. A real challenge was this checkbox with three states: **selected**, **not selected** and **neither** (i.e. don't care). You see this type of checkbox in configuration GUIs. My challenge was that it should support any Look & Feel.

After scratching my head for a while, I noticed that the ordinary JCheckBox also had three states: selected/unarmed, selected/armed, deselected/unarmed. The selected/armed state looked exactly like the "don't care" state that I wanted. [There is actually a fourth state: deselected/armed, but I did not find a use for that] The armed state is set when you have pressed the mouse button on the control, but have not released it yet.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

It took me a while to get it working, and it was a long time ago. Fortunately, since I love useful comments, I wrote a comment describing the steps needed to get this to work:

You have to overwite addMouseListener() to do nothing
You have to add a mouse event on mousePressed by calling super.addMouseListener()
You have to replace the UIActionMap for the keyboard event "pressed" with your own one.
You have to remove the UIActionMap for the keyboard event "released".
You have to grab focus when the next state is entered, otherwise clicking on the component won't get the focus.
You have to make a TristateDecorator as a button model that wraps the original button model and does state management.

I also wanted to use decent enumerated types, rather than just an int, so used my "static inner class with private constructor" trick.

Let's look at the code:

```
import javax.swing.*;
import javax.swing.event.ChangeListener;
import javax.swing.plaf.ActionMapUIResource;
import java.awt.event.*;

/**
 * Maintenance tip - There were some tricks to getting this code
 * working:
 *
 * 1. You have to overwite addMouseListener() to do nothing
 * 2. You have to add a mouse event on mousePressed by calling
 * super.addMouseListener()
 * 3. You have to replace the UIActionMap for the keyboard event
 * "pressed" with your own one.
 * 4. You have to remove the UIActionMap for the keyboard event
 * "released".
 * 5. You have to grab focus when the next state is entered,
 * otherwise clicking on the component won't get the focus.
 * 6. You have to make a TristateDecorator as a button model that
 * wraps the original button model and does state management.
 */
public class TristateCheckBox extends JCheckBox {
  /** This is a type-safe enumerated type */
  public static class State { private State() { } }
  public static final State NOT_SELECTED = new State();
  public static final State SELECTED = new State();
  public static final State DONT_CARE = new State();

  private final TristateDecorator model;

  public TristateCheckBox(String text, Icon icon, State initial){
    super(text, icon);
    // Add a listener for when the mouse is pressed
    super.addMouseListener(new MouseAdapter() {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    public void mousePressed(MouseEvent e) {
      grabFocus();
      model.nextState();
    }
  });
  // Reset the keyboard action map
  ActionMap map = new ActionMapUIResource();
  map.put("pressed", new AbstractAction() {
    public void actionPerformed(ActionEvent e) {
      grabFocus();
      model.nextState();
    }
  });
  map.put("released", null);
  SwingUtilities.replaceUIActionMap(this, map);
  // set the model to the adapted model
  model = new TristateDecorator(getModel());
  setModel(model);
  setState(initial);
}
public TristateCheckBox(String text, State initial) {
  this(text, null, initial);
}
public TristateCheckBox(String text) {
  this(text, DONT_CARE);
}
public TristateCheckBox() {
  this(null);
}

/** No one may add mouse listeners, not even Swing! */
public void addMouseListener(MouseListener l) { }
/**
 * Set the new state to either SELECTED, NOT_SELECTED or
 * DONT_CARE.  If state == null, it is treated as DONT_CARE.
 */
public void setState(State state) { model.setState(state); }
/** Return the current state, which is determined by the
 * selection status of the model. */
public State getState() { return model.getState(); }
/**
 * Exactly which Design Pattern is this?  Is it an Adapter,
 * a Proxy or a Decorator?  In this case, my vote lies with the
 * Decorator, because we are extending functionality and
 * "decorating" the original model with a more powerful model.
 */
private class TristateDecorator implements ButtonModel {
  private final ButtonModel other;
  private TristateDecorator(ButtonModel other) {
    this.other = other;
  }
  private void setState(State state) {
    if (state == NOT_SELECTED) {
      other.setArmed(false);
      setPressed(false);
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      setSelected(false);
    } else if (state == SELECTED) {
      other.setArmed(false);
      setPressed(false);
      setSelected(true);
    } else {  // either "null" or DONT_CARE
      other.setArmed(true);
      setPressed(true);
      setSelected(true);
    }
  }
  /**
   * The current state is embedded in the selection / armed
   * state of the model.
   *
   * We return the SELECTED state when the checkbox is selected
   * but not armed, DONT_CARE state when the checkbox is
   * selected and armed (grey) and NOT_SELECTED when the
   * checkbox is deselected.
   */
  private State getState() {
    if (isSelected() && !isArmed()) {
      // normal black tick
      return SELECTED;
    } else if (isSelected() && isArmed()) {
      // don't care grey tick
      return DONT_CARE;
    } else {
      // normal deselected
      return NOT_SELECTED;
    }
  }
  /** We rotate between NOT_SELECTED, SELECTED and DONT_CARE.*/
  private void nextState() {
    State current = getState();
    if (current == NOT_SELECTED) {
      setState(SELECTED);
    } else if (current == SELECTED) {
      setState(DONT_CARE);
    } else if (current == DONT_CARE) {
      setState(NOT_SELECTED);
    }
  }
  /** Filter: No one may change the armed status except us. */
  public void setArmed(boolean b) {
  }
  /** We disable focusing on the component when it is not
   * enabled. */
  public void setEnabled(boolean b) {
    setFocusable(b);
    other.setEnabled(b);
  }
  /** All these methods simply delegate to the "other" model
   * that is being decorated. */
  public boolean isArmed() { return other.isArmed(); }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    public boolean isSelected() { return other.isSelected(); }
    public boolean isEnabled() { return other.isEnabled(); }
    public boolean isPressed() { return other.isPressed(); }
    public boolean isRollover() { return other.isRollover(); }
    public void setSelected(boolean b) { other.setSelected(b); }
    public void setPressed(boolean b) { other.setPressed(b); }
    public void setRollover(boolean b) { other.setRollover(b); }
    public void setMnemonic(int key) { other.setMnemonic(key); }
    public int getMnemonic() { return other.getMnemonic(); }
    public void setActionCommand(String s) {
      other.setActionCommand(s);
    }
    public String getActionCommand() {
      return other.getActionCommand();
    }
    public void setGroup(ButtonGroup group) {
      other.setGroup(group);
    }
    public void addActionListener(ActionListener l) {
      other.addActionListener(l);
    }
    public void removeActionListener(ActionListener l) {
      other.removeActionListener(l);
    }
    public void addItemListener(ItemListener l) {
      other.addItemListener(l);
    }
    public void removeItemListener(ItemListener l) {
      other.removeItemListener(l);
    }
    public void addChangeListener(ChangeListener l) {
      other.addChangeListener(l);
    }
    public void removeChangeListener(ChangeListener l) {
      other.removeChangeListener(l);
    }
    public Object[] getSelectedObjects() {
      return other.getSelectedObjects();
    }
  }
}
```

Here is some sample code that uses the TristateCheckBox:

```
import javax.swing.*;
import java.awt.*;

public class TristateCheckBoxTest {
  public static void main(String args[]) throws Exception {
    JFrame frame = new JFrame("TristateCheckBoxTest");
    frame.getContentPane().setLayout(new GridLayout(0, 1, 5, 5));
    final TristateCheckBox swingBox = new TristateCheckBox(
        "Testing the tristate checkbox");
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
    swingBox.setMnemonic('T');
    frame.getContentPane().add(swingBox);
    frame.getContentPane().add(new JCheckBox(
        "The normal checkbox"));
    UIManager.setLookAndFeel(
        UIManager.getSystemLookAndFeelClassName());
    final TristateCheckBox winBox = new TristateCheckBox(
        "Testing the tristate checkbox",
        TristateCheckBox.SELECTED);
    frame.getContentPane().add(winBox);
    final JCheckBox winNormal = new JCheckBox(
        "The normal checkbox");
    frame.getContentPane().add(winNormal);
    // wait for 3 seconds, then enable all check boxes
    new Thread() { {start();}
      public void run() {
        try {
          winBox.setEnabled(false);
          winNormal.setEnabled(false);
          Thread.sleep(3000);
          winBox.setEnabled(true);
          winNormal.setEnabled(true);
        } catch (InterruptedException ex) { }
      }
    };
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.pack();
    frame.show();
  }
}
```

Tri it out!

Kind regards

Heinz

Maximum Solutions is not only known for **The Java(tm) Specialists' Newsletter**, but also for presenting the best training on Design Patterns that you will find anywhere :-) Please contact us so that we can put together a proposal for you, with no obligation!

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

..... [Issue 083] - End of Year Puzzle

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 83rd edition of **The Java(tm) Specialists' Newsletter**. The year 2003 is now almost over, and before we say "goodbye" completely, I thought to slip in one more quick newsletter to test your Java skills :-)

Some good news: Sun has made available a beta version of JDK 1.5. This has been negotiated by my friends over at JavaLobby, and all you have to do is register for free at their website. The URL to go to is http://www.javalobby.org/members/j2se15.jsp.

Translations: We started converting **The Java(tm) Specialists' Newsletter** into some other languages besides English [although someone suggested that we should translate it into English as well ;-] We now have translations into Portuguese (Rafael Steil and Vanessa Sabino), Zulu (Mondli Mabaso), Polish (Daniel Kwiecinski), Romanian (Dikran Seropian), Catalan (Horaci Macias), Spanish (Horaci Macias) and German (Heinz Kabutz). The translations are available on our archive page.

End of Year Puzzle

This time of year is called the *silly season* in South Africa. In Cape Town, as soon as December starts, and all the holiday makers arrive, the road works spring up all over the city. This exacerbates the traffic situation and gridlocks the roads. Maybe it takes 11 months to decide on the contractor, and then the work has to be finished before the end of the year?

So, seeing that it is "silly season", here is a little Java puzzle for your amusement, based on an idea that was sent to me by Steve Mabbort and Ben Halton. It is almost like a question you would find in the Sun Certified Java Programmer examination...

What is the output of the following Java snippet?

```
public class Puzzle1 {
  public int test() {
    int i = 1;
    try {
      return i;
    } finally {
      i = 2;
    }
  }
  public static void main(String[] args) {
    Puzzle1 p =  new Puzzle1();
    System.out.println(p.test());
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

There are several possibilities, so let's make a multiple choice out of it. To make it interesting, why don't you click on the [Select] link next to the answer that you would think to be correct? I will then tally the answers and post the result in the next newsletter.

The code does not compile [Select]
The code compiles, but throws an exception [Select]
The code prints "0" [Select]
The code prints "1" [Select]
The code prints "2" [Select]
None of the above [Select]
All of the above [Select]

After you have sent me your answer, you should try it out and see whether you were correct.

Let's change the puzzle a bit. Now it is not so obvious anymore.

```
public class Puzzle2 {
  private int i = 1;
  public int test () {
    try {
      return i;
    } finally {
      i = 2;
    }
  }
  public static void main(String[] args) {
    Puzzle2 p = new Puzzle2();
    System.out.println(p.test());
  }
}
```

The code does not compile [Select]
The code compiles, but throws an exception [Select]
The code prints "0" [Select]
The code prints "1" [Select]
The code prints "2" [Select]
None of the above [Select]
All of the above [Select]

To understand how and why this is so, I would like to suggest that you disassemble the java classes and read the byte code. You can do that with the command:

```
javap -c Puzzle1 Puzzle2
```

That's all for this year, see you again in 2004 :-) I wish you a prosperous 2004, both physically and spiritually :-)

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 083b] - End of Year Puzzle Follow-up

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Hi again,

A great number of subscribers are on holiday at the moment, but to those that are not, thanks for sending your answer to the puzzle. Approximately 70% of respondents had the correct answer, which either means that the puzzle was way too easy, or that the correct answer was also the most obvious one.

As a short follow-up, here is a variation of the puzzle for your amusement.

What is the output of the following Java snippet?

```
public class Puzzle3 {
  public int test() {
    int i = 0;
    try {
      i++;
      return i;
    } finally {
      i--;
      return i;
    }
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

```
  public static void main(String[] args) {
    Puzzle3 p =  new Puzzle3();
     System.out.println(p.test());
  }
}
```

And here is the multiple choice again:

The code does not compile [Select]
The code compiles, but throws an exception [Select]
The code prints "0" [Select]
The code prints "1" [Select]
The code prints "2" [Select]
None of the above [Select]
All of the above [Select]

I expect the split between correct/incorrect to be similar to the previous "puzzles", since the question is so similar.

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 084] - Ego Tripping with Webservices

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 84th edition of **The Java(tm) Specialists' Newsletter**, and welcome to the year 2004 A.D. Many years ago, Cape Town attempted to host the

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Olympics, and that was supposed to happen this year. Unfortunately, or perhaps fortunately, Athens won the bid. We are this year celebrating 10 years of democracy, and today, 10 years ago, my brother tied the knot, two weeks before me. Congrats, Rudolf and Karin :-)

January 2004 is a significant month. It is the first month since November 2001 that I did not publish a newsletter. I appreciate all the emails that I have been getting from you, urging me to continue, for example:

long time no newsletter :(

I just wanted to let you know that I enjoyed reading your
newsletter 82 in German.

and this one:

Hi,

What happened u look very busy?
I m waiting 4 ur next issue.
;-)

Thank you so much for these little notes, they are appreciated :-) I was quite busy with an important project that had to be finished by the 31st of January (or no payment), so whilst the rest of the world was celebrating, I had my nose against the grindstone, typing Java code at neck breaking speed.

Before we get to today's newsletter, I would like to feedback on newsletter 83 and the follow-up. I omitted to mention that Clark Updike sent me a similar code snippet to what I used in Puzzle 3. Thanks, Clark :-) Thanks to all my readers for your eager participation in the puzzle. Here are the results:

Puzzle 1: 254 out of 412 responses were *correct*, i.e. 62%
Puzzle 2: 195 out of 259 responses were *correct*, i.e. 75%
Puzzle 3: 217 out of 319 responses were *correct*, i.e. 68%

I interpret the figures as demonstrating that the readers of **The Java(tm) Specialists' Newsletter** are the smartest of any Java newsletter :-)

Ego Tripping with Webservices

These last two weeks seemed like the longest since I was battling to understand the Visitor pattern. One of my customers asked me to investigate how we could plug into Webservices and supply our own transport for the XML messages, since HTTP was not appropriate for their application.

I must admit that I viewed Webservices with skepticism, shared by many others. How many useful Webservices are we going to see in the next few years, or is it all just Silicon Snake Oil (great book, btw)? My frustrations were compounded by battling to get any communication with REAL Webservices going. Believe it or not,

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

but I struggle with new technologies. I remember when EJBs first appeared on the scene and how long it took me to deploy my first CMP bean!

The first implementation we looked at was Sun's Java Web Services Developer's Kit. The jax-rpc implementation worked most of the time, but I lost much hair (and time) when it didn't.

I am by no means an expert on Webservices. My total real experience is about two weeks. However, after starting to use the Apache Axis implementation, I become excited at the possibilities of integrating with world-famous websites from within my Java programs.

A Mathematical Webservice

First off, I will show you a Java Webservice that you can integrate with:

```
public class Fibonacci {
  public int calculate(int n) {
    if (n < 0)
      throw new IllegalArgumentException("number must be >= 0");
    switch (n) {
      case 0:
        return 0;
      case 1:
        return 1;
      default:
        return calculate(n - 2) + calculate(n - 1);
    }
  }
}
```

The Fibonacci calculator is a favourite exercise that I use to break the spirit of overconfident Java students. Talking of students, here is a quick commercial break: *Get trained by expert programmers. Come to Maximum Solutions and learn how to write excellent, efficient, good Java code. Please contact us by simply replying to this email.* The reason this code causes problems is because of its recursive nature. Even working out fibonacci(100) would take a long long time using this algorithm.

How do we convert this Java class to a Webservice? Do we compile it? Do we add an interface, and write a WSDL (Web Services Description Language) description for it?

No, it is *much* easier. Start by downloading Apache Axis v 1.1. Install Axis and Tomcat 4.1.x. Next take the directory axis in webapps (under the axis install) and copy it into the webapps directory under Tomcat. Start up Tomcat. Axis should now be working. Go to http://localhost:8080/axis and *validate* your setup. If it did not work, please follow the documentation that comes with Axis.

Now comes the cool part. Take Fibonacci.java (not the class file), copy it to the webapps/axis directory and rename it to Fibonacci.jws. ***Congratulations!*** You have written your first Webservice. That is really how easy it is. You can verify that it worked by going to http://localhost:8080/axis/Fibonacci.jws. The standard webpage

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

gives you a link to the WSDL file describing the interface for that class in a language that everybody can understand (except humans).

Accessing the Webservice

The client to the Webservice is a bit more difficult (an easier version comes later):

```
import org.apache.axis.client.*;

public class FibClient {
  public static void main(String[] args) throws Exception {
    String endpoint = "http://localhost:8080/axis/Fibonacci.jws";
    Service service = new Service();
    Call call = (Call) service.createCall();
    call.setTargetEndpointAddress(endpoint);
    call.setOperationName("calculate");
    Integer ret = (Integer) call.invoke(new Object[]{new Integer(args[0])});
    System.out.println("Got result : " + ret);
  }
}
```

You need to have your classpath set up correctly. I assume that regular readers of **The Java(tm) Specialists' Newsletter** would also be aware of the system property java.ext.dirs that you can point at a directory of jar files, instead of listing them all separately in your classpath. For example, Axis is installed on c:\java\axis-1_1 on my disk, and the compiled classes are in a subdirectory called classes. So once I have compile my FibClient, I can run it like this:

```
java -Djava.ext.dirs=c:\java\axis-1_1\lib -classpath classes FibClient 0
Got result : 0
java -Djava.ext.dirs=c:\java\axis-1_1\lib -classpath classes FibClient 1
Got result : 1
java -Djava.ext.dirs=c:\java\axis-1_1\lib -classpath classes FibClient 2
Got result : 1
java -Djava.ext.dirs=c:\java\axis-1_1\lib -classpath classes FibClient 3
Got result : 2
java -Djava.ext.dirs=c:\java\axis-1_1\lib -classpath classes FibClient 4
Got result : 3
java -Djava.ext.dirs=c:\java\axis-1_1\lib -classpath classes FibClient 5
Got result : 5
java -Djava.ext.dirs=c:\java\axis-1_1\lib -classpath classes FibClient 6
Got result : 8
java -Djava.ext.dirs=c:\java\axis-1_1\lib -classpath classes FibClient 30
Got result : 832040
```

Making the Client Code Simpler

IBM is a great company. They write excellent open source, free software, that we can use to produce good software. One of the wonderful tools that they wrote is WSDL2Java. You can invoke it on the Webservice that we have just written:

```
java -Djava.ext.dirs=c:\java\axis-1_1\lib org.apache.axis.wsdl.WSDL2Java
  http://localhost:8080/axis/Fibonacci.jws?wsdl -o src -p com.maxoft.fib.client
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

This generates some files that we can use to make our client easier to write. The package that the files are in is com.maxoft.fib.client, so that is where I will add my new FibClient class:

```
package com.maxoft.fib.client;

public class FibClient2 {
  public static void main(String[] args) throws Exception {
    FibonacciService service = new FibonacciServiceLocator();
    Fibonacci fib = service.getFibonacci();
    for (int i=0; i<10; i++) {
      System.out.println("fib(" + i + ")=" + fib.calculate(i));
    }
  }
}
```

And that is all you have to do. Run the code like this:

```
java -Djava.ext.dirs=lib -classpath classes com.maxoft.fib.client.FibClient2
fib(0)=0
fib(1)=1
fib(2)=1
fib(3)=2
fib(4)=3
fib(5)=5
fib(6)=8
fib(7)=13
fib(8)=21
fib(9)=34
```

And Now, the Ego Trip

Fame is when you walk into a room and there are more people that know you than that you know. My father-in-law was the best soccer player in South Africa in his days of glory. For 20 years afterwards, people would greet him in the street and say: *Remember in 1968 when we celebrated your victory together?* Of course he did not, but he did not want to disappoint his fans either.

The question is: How famous are *you*? Depending on your name, that may be hard to determine. My name is easy. I am the only "Heinz Kabutz" alive on this planet. My friend John Green probably has another John Green living in the same street. So, the results of a Google search must not be taken too seriously.

Google has published a Webservice interface that we can use to do searches. The WSDL is available at http://api.google.com/GoogleSearch.wsdl. Google makes this available to the general public, so there are restrictions. For example, you have to register at their website to obtain a license key, that you send along with every request. You can only do 1000 requests per day, and there are other restrictions. Please do read the documentation on Google's website. I would like to continue

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

doing EgoSurfing on Google through my program, so please do not abuse their goodwill ;-)

In addition to the WSDL, Google makes available a Java API Facade. However, since I was trying to learn how to interact with a Webservice, I used Axis to speak to Google. It was a lot easier than I thought it would be. First off, it is a good idea to generate Java classes from the WSDL:

```
java -Djava.ext.dirs=c:\java\axis-1_1\lib org.apache.axis.wsdl.WSDL2Java
  http://api.google.com/GoogleSearch.wsdl -o src
```

This creates the following files in the package GoogleSearch:

```
DirectoryCategory.java
GoogleSearchBindingStub.java
GoogleSearchPort.java
GoogleSearchResult.java
GoogleSearchService.java
GoogleSearchServiceLocator.java
ResultElement.java
```

The stage is now set for our EgoTrip client. I am using the generated Google classes to do a search for famous names:

```
package com.maxoft.webservices;

import GoogleSearch.*;

import javax.xml.rpc.ServiceException;
import java.rmi.RemoteException;
import java.util.*;

public class EgoTrip {
  private final GoogleSearchPort port;
  private final String googleKey;

  public EgoTrip(String googleKey) throws ServiceException {
    this.googleKey = googleKey;
    GoogleSearchService service = new GoogleSearchServiceLocator();
    port = service.getGoogleSearchPort();
  }

  private int measureFame(String name) throws RemoteException {
    GoogleSearchResult result = port.doGoogleSearch(
       googleKey, name, 0, 10, false, "", false, "", "UTF-8", "UTF-8");
    return result.getEstimatedTotalResultsCount();
  }

  public static void main(String[] args) throws Exception {
    String[] famousPeople = {
      "Bruce Eckel", // famous Java author
      "John Vlissides", // famous Design Pattern author
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
      "Cay Horstmann", // famous Java author
      "Erich Gamma", // famous Design Pattern author
      "Martin Fowler", // famous author
      "J•rn Schmidt-Dumont", // my sales manager
      "Heinz Kabutz", // me
      "Pieter Kritzinger", // my prof at university
      "Helene Kabutz", // my wife
      "Marten Kabutz", // my older brother
      "Rudolf Kabutz", // my oldest brother
      "Bettina Kabutz", // my sister
      "Bjarne Stroustrup", // famous C++ inventor
      "Mark Shuttleworth", // famous Thawte founder
      "Bill Gates", // apparently wrote some software
      "Costa Kytides", // #14 in the world street-luger
      "Anesti Kytides", // stand-up skate-boarder
      "John Green", // friend from university
    };
    Set people = new TreeSet();

    EgoTrip et = new EgoTrip(args[0]);
    for (int i = 0; i < famousPeople.length; i++) {
      System.out.println("Egotripping " + famousPeople[i]);
      int fame = et.measureFame(famousPeople[i]);
      people.add(new Person(fame, famousPeople[i]));
    }

    System.out.println();
    System.out.println("Hall of Snake Oil Fame");
    for (Iterator it = people.iterator(); it.hasNext();) {
      System.out.println(it.next());
    }
  }

  private static class Person implements Comparable {
    private final int fame;
    private final String name;

    public Person(int fame, String name) {
      this.fame = fame;
      this.name = name;
    }

    public int compareTo(Object o) {
      Person other = (Person) o;
      int result = other.fame - fame;
      if (result == 0) {
        return name.compareTo(other.name);
      }
      return result;
    }

    public String toString() {
      return name + " (" + fame + ")";
    }
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

}

When you run this program, you have to give it your Google key that you get when you register. You can register on Google Web APIs website. You can then run this program yourself:

```
java -Djava.ext.dirs=lib -classpath classes com.maxoft.webservices.EgoTrip
YOUR_GOOGLE_LICENSE_KEY
```

By the time you run this program, the newsletter should be on the website, so perhaps Google would've done its work indexing it? If so, the numbers could be different. Infact, they will probably be different because they are just an estimate. Sorry Bill Gates, you are not at the top, my friend John Green is more famous:

```
Egotripping Bruce Eckel
Egotripping John Vlissides
Egotripping Cay Horstmann
Egotripping Erich Gamma
Egotripping Martin Fowler
Egotripping J•rn Schmidt-Dumont
Egotripping Heinz Kabutz
Egotripping Pieter Kritzinger
Egotripping Helene Kabutz
Egotripping Marten Kabutz
Egotripping Rudolf Kabutz
Egotripping Bettina Kabutz
Egotripping Bjarne Stroustrup
Egotripping Mark Shuttleworth
Egotripping Bill Gates
Egotripping Costa Kytides
Egotripping Anesti Kytides
Egotripping John Green

Hall of Snake Oil Fame
John Green (3360000)
Bill Gates (1630000)
Martin Fowler (207000)
Bruce Eckel (29300)
Bjarne Stroustrup (22900)
Erich Gamma (21600)
Mark Shuttleworth (16000)
John Vlissides (14000)
Cay Horstmann (7880)
Heinz Kabutz (874)
Pieter Kritzinger (846)
Rudolf Kabutz (36)
J•rn Schmidt-Dumont (35)
Anesti Kytides (18)
Costa Kytides (16)
Helene Kabutz (11)
Bettina Kabutz (9)
Marten Kabutz (6)
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

At least I appear more often than my professor from the University of Cape Town ;-)

Thanks for reading this newsletter :-)

Kind regards

Heinz

Copyright 2000-2003 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 085] - Book Review: Pragmatic Programmer

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 85th edition of **The Java(tm) Specialists' Newsletter**. Isn't it amazing how quickly our bodies can deteriorate when we work as programmers? When I was at school, I used to go spearfishing about once a week, and I was under the impression that I was unfit. Ha! I had NO idea what unfit really meant. So, here is a challenge for all of us professional Java programmers: Get off your comfortable chair, and do 50 pushups. Yes, you can use both hands. If your boss walks in while you do that, tell him that **The Java(tm) Specialists' Newsletter** said you should do this. Back in the days when I thought I was unfit, I used to do that in 20 seconds.

I would be interested to hear how many pushups you could do without taking a break, and what the reaction from your co-workers was :-)

For this week's newsletter, I am again reviewing a book, this time "The Pragmatic Programmer - from journeyman to master" by Andrew Hunt and David Thomas. An interesting point of note: This book is the #13 bestseller via Amazon in South Africa. I think that a large portion of all books in South Africa are bought via Amazon, so that is quite an achievement!

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Book Review: The Pragmatic Programmer

The Pragmatic Programmer was published 15 days after "eXtreme Programming Explained: Embrace Change" by Kent Beck, by the same publishing house. If they had appeared a long time apart, I would have said that they borrowed heavily from each other. That said, I found **The Pragmatic Programmer** to be more useful.

Dictionary.com has several definitions of "pragmatic", the most appropriate being:

Dealing or concerned with facts or actual occurrences; practical; and
guided by practical experience and observation rather than theory

Academics are known to be the opposite of pragmatic. It is not always good to be pragmatic, since it can limit your dreaming. There are few "pragmatic" doctorates of computer science research theses. We have to be balanced in our approach. I know of one educational institution in Cape Town that produces programmers who think that *any* algorithm can be solved by Delphi. When I asked them: Can your marvelous language solve the halting problem algorithm? Can it solve NP-Complete algorithms in polynomial time? The answer was: "Of course". That would be an example of too much pragmatism and not enough theory.

Personally, I prefer a balanced approach. Go to university and learn as much theory as you possibly can. Do at least a Masters of Computer Science. Then keep the theory in the back of your mind, and concentrate on the pragmatic issues surrounding real-life software development. This is where a book like The Pragmatic Programmer will be of tremendous benefit.

The book has lots of tips sprinkled throughout the book. For example:

Tip 4: Don't live with Broken Windows
Tip 8: Invest Regularly in Your Knowledge Portfolio, e.g.
Learn at least one new language every year.
Read a technical book each quarter.
Read nontechnical books too.
Take classes. (e.g. Design Patterns Course)
Participate in local user groups.
Experiment with different environments.
Stay current.

Tip 11: *DRY* - *D*on't *R*epeat *Y*ourself
Tip 29: Write Code that Writes Code

The book obviously expounds on these tips with lots of examples from real experiences.

Law of Demeter for functions

This law was taken from a paper published in 1989 on "Assuring good style for object-oriented programs". It was nicely described in the book using a C++ class, here it is using Java:

```
public class Demeter {        // The Law of Demeter for functions
  private A a;                // states that any method of an
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  private int func() { ... }   // object should call only methods
  public void example (B b) {  // belonging to:
    int f = func(); // <--------- itself

    b.invert(); // <------------- any parameters that were passed
                          // in to the method

    a = new A();
    a.setActive(); // <---------- any objects it created
  }
}
```

After each section there are Exercises with model solutions. Let's have a look at the exercises for the Law of Demeter. Are these method calls allowed according to the *Law of Demeter*?

```
public void showBalance(BankAccount acct) {
  Money amt = acct.getBalance();
  printToScreen(amt.printFormat());
}
```

and

```
public class Colada {
  private Blender myBlender;
  private Vector myStuff;
  public Colada() {
    myBlender = new Blender();
    myStuff = new Vector();
  }
  private void doSomething() {
    myBlender.addIngredients(myStuff.elements());
  }
}
```

I would implore you to spend some time thinking about this and applying the rules in the Demeter class before looking at the answer.

Here is the answer from the book for the first question: *The method showBalance() breaks the law of demeter. The variable acct is passed in as a parameter, so the getBalance() call is allowed. Calling amt.printFormat(), however, is not. We don't "own" amt and it wasn't passed to us. We could eliminate showBalance's coupling to Money with something like this:*

```
public void showBalance(BankAccount acct) {
  acct.printBalance();
}
```

I must admit that I failed this one miserably. According to the book, research has shown that classes designed in this way will also tend to have fewer errors. There is a website dedicated to the Law of Demeter at http://www.ccs.neu.edu/research/demeter. If you read the Greek, it actually should

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

be the Law of Dimitra, the Greek goddess of agriculture. It is related to Aspect-Oriented and Adaptive programming.

In the second question, since Colada creates and owns both myBlender and myStuff, the calls to addIngredients and elements are allowed.

I have started considering the Law of Demeter when coding my methods. Sometimes I adhere to it, other times not. It is not always easy to change your bad habits, as people who bite their fingernails will tell you.

Automation

Another point that the book emphasises is automation. With the marvelous Ant tool, we really do not have an excuse anymore. Our build process should be completely automated as well as most of our testing. The more you automate, the better chance of you producing systems that are regularly built and tested. Through regression testing, you will pick up problems much quicker.

The book does not mention Ant, since that was developed after the book was published. We have the advantage of being able to set up our build script to run with Ant every night, and then email us the test results. There is a nice product called Anthill that will help you automate the Ant builds and create a project website for your product.

Assertions

Tip 33: If it can't happen, use assertions to ensure that it won't.

The book suggests writing lots of assertion code, and leaving it in the final build that goes to customers. This is the only place I have seen this statement. I have always thought something like this:

Assertions add some overhead to code. Because they check for things that should never happen, they'll get triggered only by a bug in the code. Once the code has been tested and shipped, they are no longer needed, and should be turned off to make the code run faster. Assertions are a debugging facility.

The book says that turning assertions off when you deliver your program to production is like crossing a high wire without a net because you once made it across in practice. They also suggest that many things that your assertions test for simply do not happen during testing, but can happen, even they they cannot, during production.

What concerns me is what happens to an uncaught AssertionError in Java. Ideally you should have a framework that catches all unhandled exceptions and all errors, and issues warnings to the system administrator. Since AssertionError falls under Error and not Exception, we have to be vigilant in handling all Throwables, not just Exceptions.

On a related note, we have Tip 32: Crash early. As soon as you know that there is a problem, crash. Don't try to continue operating with a half-working system. Rather crash. There are different ways that you can crash (not trash). For example, you could send an email to the administrator, or put an entry in a log file. You could end

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

the program completely. However, you **have** to let someone know that there is a problem. This reminds me of a program that I saw written using J2EE. This program had its own log file, and every time an error occurred, it would write to its log file. No one knew where this log file was located. There were other issues, such as that the log file could grow indefinitely. This could potentially have filled up the hard disk, thus causing more errors. The only way that you could see something was wrong was that the data was not filled in on the screen. No error message on the console, nothing, just deadly quiet (except for the hidden log file). Rather crash early than hide problems.

I can heartily recommend this book as being the best practical advice book on how to write real computer programs in today's environment. I have learnt a lot from "The Pragmatic Programmer" and you will too. Another good book on a similar topic is Code Complete by Steve McConnell.

And now, for some other books...

If I were to dedicate one whole newsletter to each good book that I own, I would be busy for a long time. I have additional books that I want to review, but don't have the time, so this section will list several books, with a short comment

My Brain is Open on the Mathematical Journeys of Paul Erd˜s by Bruce Schechter. A fascinating short book about one of the greatest mathematicians of the last century.

Head First Java is a nice book for people who want to learn Java, i.e. not for your typical The Java(tm) Specialists' Newsletter subscriber ;-) It has lots of pictures and diagrams and looks like a schoolbook for 12 year olds (in a positive way!). Despite its comical appearance, I have found that the topics are explained accurately. Full of information, I would recommend this book as a first book for anyone wanting to learn Java from scratch.

Java Enterprise Best Practices contains lots of tips and tricks for topics on EJB, JDBC, RMI, XML, Servlets, JavaMail, etc. It is definitely written for us Java Specialists, rather than beginners. I have applied a few of the ideas and they work well.

Patterns of Enterprise Application Architecture by Martin Fowler. Martin is the best technical writer I know. He writes with a style that is entertaining, yet not irritating (difficult balance). This is an excellent book focusing on enterprise systems. Another great book by Martin Fowler is of course Refactoring, which should be required reading for all Object-Oriented developers.

Java Extreme Programming Cookbook by Eric Burke and Brian Coyner will help you set up an automated system that will make it possible to apply the principles in The Pragmatic Programmer.

That's all for this week :-) Happy reading, and don't forget to do those pushups...

Kind regards

Heinz

Copyright 2000-2004 Maximum Solutions, South Africa

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 086] - Initialising Fields before Superconstructor call

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Welcome to the 86th edition of **The Java(tm) Specialists' Newsletter**. My last newsletter caused some interesting reactions. A client told me that her programmers got quite excited talking about pushups, then decided that this could be better discussed at the local pub. The rest of the afternoon was spent drinking beers. Not quite the reaction I had hoped for, but at least I got them th(dr)inking!

The South African government is trying to promote education within our country. One of the ideas is that of learnerships, which are supposed to work like an apprenticeship. This is for South African citizens only, unfortunately. I don't know how it all works, you can have a look on the ISETT SETA website for more information. I am considering the option of running learnerships at Maximum Solutions, please have a look at my website if this interests you.

Initialising Fields before Superconstructor call

One of the biggest surprises that I encountered when I first learned Java was when Bruce Eckel pointed out in his book that when a method is called from the superconstructor, the most derived method is invoked. The fields in the subclass are not yet initialised, which can lead to nasty bugs. I have written about this phenomenon before, and my message was always: "Don't call non-private methods from your constructors!"

But what do you do if you are working within a framework, and your superclass has this behaviour? In a recent project, I wanted to have the class take as a parameter a business object class, and depending on the type, create a different GUI Form, via some FormFactory. It looked something like this:

```
import javax.swing.*;

public abstract class FormView1 {
  private final JComponent mainUIComponent;
  public FormView1(JFrame owner) {
    // do some stuff...
    mainUIComponent = makeUI();
  }
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  public abstract JComponent makeUI();
  public JComponent getMainUIComponent() {
    return mainUIComponent;
  }
}


import javax.swing.*;

public class CustomerView1 extends FormView1 {
  private final Integer type;
  public CustomerView1(JFrame owner, int type) {
    super(owner);
    this.type = new Integer(type);
  }
  public JComponent makeUI() {
    switch (type.intValue()) {
      case 0:
        return new JTextArea();
      case 1:
        return new JTextField();
      default:
        return new JComboBox();
    }
  }
  public static void main(String[] args) {
    CustomerView1 view1 = new CustomerView1(null, 1);
    System.out.println(view1.getMainUIComponent().getClass());
  }
}
```

What happens when you run that code? To my newsletter readers, the answer will be obvious: NullPointerException! The sequence of calls is as follows:

```
java.lang.NullPointerException
  at CustomerView1.makeUI(CustomerView1.java:13)
  at FormView1.<init>(FormView1.java:9)
  at CustomerView1.<init>(CustomerView1.java:8)
  at CustomerView1.main(CustomerView1.java:24)
```

main(String[]) constructs a CustomerView1
Before CustomerView1 has initialised any variables, it calls super(owner) [note that the `type` field has not been initialised yet]
FormView1 makes a call to makeUI() which uses the type field before it has been initialised.

Ok, this is nothing new. People have been writing about this problem for many years.

A good solution to this problem would be to change the framework. For example, you could change the FormView to take parameters in the constructor, which it then passes to the makeUI() method, like so:

```
import javax.swing.*;

public abstract class FormView2 {
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
  private final JComponent mainUIComponent;
  public FormView2(JFrame owner, Object subclassParameters) {
   // do some stuff...
   mainUIComponent = makeUI(subclassParameters);
  }
  public abstract JComponent makeUI(Object subclassParameters);
  public JComponent getMainUIComponent() {
    return mainUIComponent;
  }
}


import javax.swing.*;

public class CustomerView2 extends FormView2 {
  private Integer type;
  public CustomerView2(JFrame owner, int type) {
    super(owner, new Integer(type));
  }
  public JComponent makeUI(Object params) {
    this.type = (Integer) params;
    switch (type.intValue()) {
      case 0:
        return new JTextArea();
      case 1:
        return new JTextField();
      default:
        return new JComboBox();
    }
  }
  public static void main(String[] args) {
    CustomerView2 view1 = new CustomerView2(null, 1);
    System.out.println(view1.getMainUIComponent().getClass());
  }
}
```

It should be obligatory to put an Object parameter in all methods that are called from a constructor in a class that is meant to be overridden. At least that way we can configure the class.

What if you cannot change the framework?

Let's assume that you cannot change the framework (or that you just do not want to). What options do you have?

The first option that springs to mind is inheritance. Instead of having one CustomerView, we make three different ones. Each of the three CustomerView classes will return a different component. However, this could potentially cause too many classes to be created. Having many classes adds to your maintenance headache.

The second option is one that I discovered this week, much to my joy. In your call to **super**(), you cannot call any non-static methods of your class, because **this** has not

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

been initialised yet. However, you *may* call static methods. So, provided that the superclass takes at least one parameter, we can write the following hack:

```
import javax.swing.*;

public class CustomerView3 extends FormView1 {
  private static final Object lock = new Object();
  private static Integer tempType;
  private Integer type;
  /** The problem with the superclass is that it makes a callback
   * to method makeUI().  We however want to set a variable in
   * this object before that method is called.  The way we do it
   * is to set a static variable, then at the beginning of the
   * makeUI() method we initialise our non-static variable. */
  public CustomerView3(JFrame owner, int type) {
    super(hackToPieces(owner, type));
  }
  private static JFrame hackToPieces(JFrame owner, int type) {
    synchronized (lock) {
      /** We want to prevent several threads overwriting the
       * tempType static variable. */
      while (tempType != null) {
        try {
          lock.wait();
        } catch (InterruptedException e) {
          // someone wants to shut us down, let's return and keep
          // the thread interrupted
          Thread.currentThread().interrupt();
          return owner;
        }
      }
      tempType = new Integer(type);
      return owner;
    }
  }
  /** We initialise the variables and set the temporary static
   * fields to null. */
  private void init() {
    synchronized (lock) {
      type = tempType;
      tempType = null;
      lock.notifyAll();
    }
  }
  public JComponent makeUI() {
    // Make sure that init() is called first.  This assumes that
    // makeUI only gets called once, by the superclass'
    // constructor.  If that is a false assumption, you will have
    // to be a bit cleverer here.  Probably test whether type
    // is null or something.  Exercise for the reader.
    init();
    switch (type.intValue()) {
      case 0: return new JTextArea();
      case 1: return new JTextField();
      default: return new JComboBox();
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
    }
  }
  public static void main(String[] args) {
    CustomerView3 view1 = new CustomerView3(null, 1);
    System.out.println(view1.getMainUIComponent().getClass());
  }
}
```

It was a great thrill to solve this problem to which I did not know a solution.

The "hack" in CustomerView3 is perfectly legitimate Java code, so I do not have a problem using it in production code, *provided that*:

the other developers are aware of what I've had to do,
the source code is generously documented with useful comments,
this approach was easier than just modifying the framework.

Have fun with this approach, and amaze your colleagues ;-)

Kind regards

Heinz

Copyright 2000-2004 Maximum Solutions, South Africa

Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

..... [Issue 086b] - Initialising Fields before Superconstructor call (Follow-up)

Author: Dr. Heinz M. Kabutz

You can subscribe from our home page: http://www.javaspecialists.co.za (which also hosts all previous issues, available free of charge :-)

Gidado-Yisa Immanuel sent me another approach that is a bit simpler to write, hence should reduce the possibility of bugs. Instead of using a static field, and worrying about concurrent access, why not rather just use a ThreadLocal? I must admit that I have never used ThreadLocals because of the problems in previous versions of Java, so that was an option I did not consider.

```
import javax.swing.*;
```

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

```
public class CustomerView4 extends FormView1 {
  private static final ThreadLocal tempType = new ThreadLocal();
  private Integer type;
  public CustomerView4(JFrame owner, int type) {
    super(hackToPieces(owner, type));
  }
  private static JFrame hackToPieces(JFrame owner, int type) {
    tempType.set(new Integer(type));
    return owner;
  }
  private void init() {
    type = (Integer) tempType.get();
  }
  public JComponent makeUI() {
    init();
    switch (type.intValue()) {
      case 0:
        return new JTextArea();
      case 1:
        return new JTextField();
      default:
        return new JComboBox();
    }
  }
  public static void main(String[] args) {
    CustomerView4 view = new CustomerView4(null, 1);
    System.out.println(view.getMainUIComponent().getClass());
  }
}
```

This certainly looks simpler than my approach, so rather use Gidado's solution than mine, even though mine is marginally faster (about 1%). And even better than Gidado's solution would be to change the framework :-)

Kind regards from

Heinz

Copyright 2000-2004 Maximum Solutions, South Africa
Reprint Rights. Copyright subsists in all the material included in this email, but you may freely share the entire email with anyone you feel may be interested, and you may reprint excerpts both online and offline provided that you acknowledge the source as follows: *This material from The Java(tm) Specialists' Newsletter by Maximum Solutions (South Africa). Please contact Maximum Solutions for more information.*

Java and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maximum Solutions is independent of Sun Microsystems, Inc.

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

24. XML

- **Question:** What is relation between JAXP and Xerces?

  Answer: JAXP is a set of interfaces and Xerces is one of JAXP implementations. JAXP plays the same role for Xerces like Oracle's JDBC Driver for Oracle.

  The Java API for XML Processing (JAXP) supports processing of XML documents using DOM, SAX, and XSLT. JAXP enables applications to parse and transform XML documents independent of a particular XML processing implementation.

  Depending on the needs of the application, developers have the flexibility to swap between XML processors (such as high performance vs. memory conservative parsers) without making application code changes.

  Thus, application and tools developers can rapidly and easily XML-enable their Java applications for e-commerce, application integration, and dynamic Web publishing. Just added into the JAXP 1.2 reference implementation is support for XML schema and an XSLT compiler (XSLTC). These new features are also available as part of the Java Web Services Developer Pack (Java WSDP) v1.0_01 and as part of the Java XML Pack Summer 02 Update Release.

- **Question:** I just looked at Java Web Services Developer Pack (Java WSDP) and found that it contains a lot of XML functionality... What is a difference between Java WSDP and Java XML Pack?

  Answer: Yes, they much in common, both XML pack and Java WSDP lets to test and deploy XML applications.

  The Java XML Pack contains individual implementations for XML and Web Services APIs developed under the Java Community ProcessSM (JCPSM).

  Java WSDP is a product that includes these and additional technologies.

  The differences are that:

  Java XML Pack lets you to pick up the exact XML technology you need, let’s name it "classical" XML set and

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Even More Tips at http://JavaA.com and http://JavaFAQ.nu every day!!!

Java WSDP is an "all-in-one" set that help you to easy develop web services and Web applications.

The Java WSDP provides Java standard implementations of existing key Web services standards including WSDL, SOAP, ebXML, and UDDI as well as important Java standard implementations for Web application development such as JavaServer Pages (JSPTM pages) and the JSP Standard Tag Library. These Java standards allow developers to send and receive SOAP messages, browse and retrieve information in UDDI and ebXML registries, and quickly build and deploy Web applications based on the latest JSP standards.

The Java Web Services Developer Pack v1.0_01 includes the following:

Java XML Pack Release which includes the following:

- Java API for XML Messaging (JAXM) v1.1_01
- Java API for XML Processing (JAXP) v1.2_01
- Java API for XML Registries (JAXR) v1.0_02
- Java API for XML-based RPC (JAX-RPC) v1.0_01
- SOAP with Attachments API for Java (SAAJ) v1.1_02
- l JavaServer Pages Standard Tag Library (JSTL) v1.0.1

Java WSDP Registry Server v1.0_02
Web Application Deployment Tool
Ant Build Tool 1.4.1
Apache Tomcat 4.1.2 container

Ł

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

25. About The Author

John Andersson is my pseudonym which I use in my newsletters and on my website JavaFAQ.nu. It will be probably long explanation why I, as well as other people, use pseudonyms. So, I skip it for "clarity" :-)

A few words about me…

I graduated from Moscow State University, physical faculty: computer design. Also I studied nuclear physics and plasma physics extensively during first four years of studying.

After University I worked in many telecommunication companies and now I am working at Ericsson AB in Stockholm as programmer in GSM area. During last years I was working mostly with Java and also High Availability UNIX Servers (where we run our SW).

I am married and have two sons. Most time I spend with them and my wife.
Also in my free time I like to "ride" my bike and twice a week I am training in sport hall.

Ł

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu

Copyright © 2005 http://JavaA.com http://JavaFAQ.nu alex@javafaq.nu
Copyright © Type Text Here Type Text Here